

Internal bus system for DFPS and units with two- or multi-dimensional programmable cell architectures, for managing large volumes of data with a high interconnection complexity

Patent number: DE19704742
Publication date: 1998-09-24
Inventor: MUENCH ROBERT (DE); VORBACH MARTIN (DE)
Applicant: PACT INF TECH GMBH (DE)
Classification:
- **International:** G06F13/38; G06F15/80
- **European:**
Application number: DE19971004742 19970211
Priority number(s): DE19971004742 19970211

Also published as:

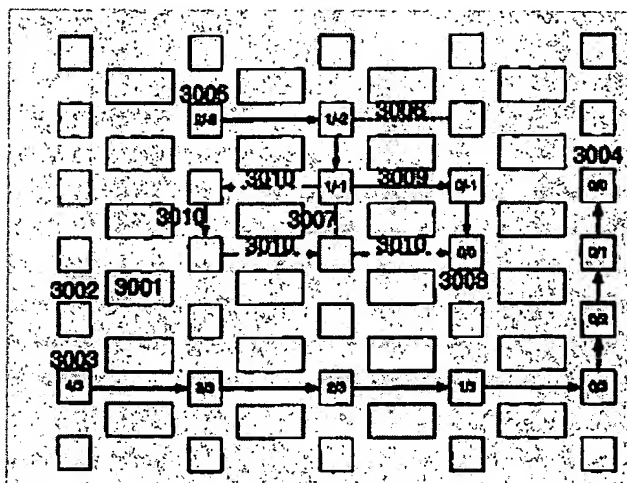


WO9835294 (A3)
WO9835294 (A2)
EP0960374 (A3)
EP0960374 (A2)
US6405299 (B1)

more >>

Abstract of DE19704742

The invention relates to a bus system for DFPS DE 16 88 A1 and to building blocks with two dimensional or multidimensional programmable cell structures (of type FPGA, DPGA or the like), characterized in that a) the bus system comprises a plurality of electrically independent bus segments disconnected by nodes, b) the nodes actively connect or disconnect bus segments, whereby b1) several bus segments are joined and connected by gates to a bus bar located inside the node or b2) connection occurs directly via circuitry elements, drivers and/or registers, c) each node has a routing table where information on the structure of the connection is stored, d) each node has a monitoring unit, which verifies independently if a connection can be established or not.





⑮ **BUNDESREPUBLIK
DEUTSCHLAND**



**DEUTSCHES
PATENTAMT**

⑫ **Offenlegungsschrift**
⑩ **DE 197 04 742 A 1**

⑤ Int. Cl.⁶:
G 06 F 13/38
G 06 F 15/80

⑲ Aktenzeichen: 197 04 742.4
⑳ Anmeldetag: 11. 2. 97
㉔ Offenlegungstag: 24. 9. 98

DE 197 04 742 A 1

㉑ **Anmelder:**
Pact Informationstechnologie GmbH, 76133
Karlsruhe, DE

㉒ **Erfinder:**
Vorbach, Martin, 76149 Karlsruhe, DE; Münch,
Robert, 76149 Karlsruhe, DE

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

- ⑤④ Internes Bussystem für DFPs, sowie Bausteinen mit zwei- oder mehrdimensionalen programmierbaren Zellstrukturen, zur Bewältigung großer Datenmengen mit hohem Vernetzungsaufwand

DE 197 04 742 A 1

Beschreibung

1 Hintergrund der Erfindung

1.1 Stand der Technik

1.1.1 ... in Systemen mit zwei- oder mehrdimensionalen programmierbaren Zellstrukturen (FPGAs, DPGAs)

In FPGAs und DPGAs befinden sich interne Bussysteme, die entweder global Anschluß an alle oder eine große Menge von Logikzellen haben oder eine lokale NextNeighbour-Verbindung aufweisen. Beiden Arten ist gemeinsam, daß es sich um direkte Verbindungen zwischen zwei oder mehreren Logikzellen handelt. Weiterhin kann immer genau ein Signal den Bus benutzen, es sei denn, daß eine Multiplexerstruktur in eine Mehrzahl von Logikzellen zusammen mit einer Steuerung konfiguriert wird.

1.1.2 ... in DFP-basierenden Systemen

In DFP-basierenden Systemen, nach DE 44 16 881 existieren die selben bereits beschriebenen Bussysteme. Zusätzlich existiert die Möglichkeit ein Bussystem zu trennen, so daß mehrere getrennt nutzbare Teilbusse entstehen.

1.1.3 ... in IO-Bussysteme nach PACT03

In PACT03 wird ein IO-Bussystem beschrieben, das eine Vielzahl Bausteininterner Bussysteme, wobei ein Baustein ein FPGA, DPGA, DFP oder ähnlicher sein kann, zusammenfaßt und aus dem Baustein herausführt. Dabei können Speicherbaugruppen oder Peripherie oder weitere Bausteine oben genannter Gattung angesteuert werden. Es existiert nur ein Adressregister, bzw. ein Adresszähler um die externen Adressen zu generieren.

1.2 Probleme

Die üblichen Bussysteme sind nicht dazu geeignet große Datenmengen in Form von byteweise oder anders gruppierten Signalen zu übertragen. Besonders wenn die Bausteine zur Berechnung von Algorithmen verwendet werden ist es erforderlich, daß eine Vielzahl von Daten(paketen) gleichzeitig zwischen den einzelnen konfigurierten Funktionsbereichen eines Bausteines übertragen werden. In der üblichen Technik muß für jeden Datenpfad, das ist die Verbindung (der Bus) zweier (oder mehrerer) Funktionsblöcke, die dieselben Daten erhalten, eine direkte Punkt zu Punkt Verbindung aufgebaut werden, die dann ausschließlich den Datenverkehr zwischen eben diesen Funktionsblöcken regelt. Es kann sich immer genau ein Datenpaket auf dem Bus befinden. Der Vernetzungsaufwand wird extrem hoch. Die Geschwindigkeit der heutigen internen Busse wird begrenzt durch die maximale Busbreite und die Laufzeit der Signale auf dem Bus. Beim in PACT03 beschriebenen IO-Bus kann pro IO-Zelle nur eine Art von Busverbindung aufgebaut werden, nämlich ausschließlich die in dem Adressregister programmierte. Es besteht keine Möglichkeit auf verschiedene Datentypen oder Datensender zu reagieren und diese mit verschiedenen externen Bausteinen zu verbinden.

1.3 Verbesserung durch die Erfindung, Aufgabe

Ziel der Erfindung ist es, ein Bussystem zu schaffen, das Daten zwischen einer Vielzahl von Funktionsblöcken übertragen kann, wobei sich mehrere Datenpakete gleichzeitig auf dem Bus befinden können. Dabei erkennt das Bussystem für verschiedene Datentypen oder Datensender automatisch die richtige Verbindung und baut diese auf. Die Einzelheiten und besondere Ausgestaltungen, sowie Merkmale des erfindungsgemäßen Bussystems sind Gegenstand der Patentansprüche.

2 Beschreibung der Erfindung

2.1 Übersicht über die Erfindung. Abstrakt

Beschrieben wird ein Bussystem, das horizontal, vertikal, diagonal oder in jeder beliebigen Lage auf einem Baustein integriert werden kann. Das Bussystem ist in eine Vielzahl von Segmenten unterteilt, wobei die Trennung der Segmente durch eine Bussteuerschaltung erfolgt. Diese Bussteuerschaltung wird Knoten genannt. Gleichzeitig übernimmt der Knoten das Routen, also das Steuern der Flußrichtung der Daten. Logikzellen oder PAE-Zellen nach PACT02 sind an die Knoten angeschlossen, sie senden und empfangen ihre Daten über die Knoten, wobei jedem Datenpaket eine Zieladresse mitgegeben werden kann. Ebenfalls ist das Bussystem in der Lage über LookUp-Tabellen, im folgenden Routingtabellen genannt, Zieladressen zu generieren, hierzu kann vor allem das in PACT04 beschriebene Verfahren dienen. Die Bussysteme sind besonders zu einem direkten Anschluß an die in PACT03 beschriebenen INPUT-/OUTPUT-CELLs geeignet. Das Auf- und Abbauen von Verbindungen, sowie die Synchronisation der Daten erfolgt automatisch. Schlägt ein Aufbau fehl, da das benötigte Bussegment momentan belegt ist, wird der Aufbau zu einem späteren Zeitpunkt wiederholt. Anhand unterschiedlicher Datentypen oder Datensender können automatisch unterschiedliche Verbindungen aufgebaut werden.

3 Detailbeschreibung der Erfindung

3.1 Das Bussystem

Auf einem Baustein befinden sich eine Vielzahl von Bussen, die horizontal, vertikal, diagonal oder in jeder beliebigen Lage auf dem Baustein angeordnet sind. Dabei sind die einzelnen Busse nicht durchgehend von einer Bausteinkante zur anderen, sondern sie sind in eine Mehrzahl von Bussegmenten unterteilt. Die Bussegmente werden durch die nachfolgend beschriebenen Knoten getrennt. Jedes Bussegment ist unabhängig von den anderen nutzbar und verschaltbar, wobei die Verschaltung von den Knoten übernommen wird. Auf dem Bussystem ist kann ein eigenes Protokoll implementiert sein, das von den Knoten verwaltet wird; ebenfalls ist denkbar, daß das Protokoll von den Zellen, die den Bus benutzen verwaltet wird und die Knoten nur passive Schalter sind.

3.2 Der Knoten

Ein Knoten dient der Verschaltung der einzelnen Bussegmente miteinander. Ebenfalls wird das Aufschalten der Zellen auf die Bussegmente von den Knoten übernommen.

In einem Knoten sind die Bussegmente aller Richtungen zusammengeführt, das bedeutet,

- in einem zweidimensionalen System werden die Busse aus 4 Richtungen Norden (N), Süden (S), Osten (E), Westen (W) auf den Knoten geführt;
- in einem dreidimensionalen System werden die Busse aus 6 Richtungen Norden (N), Süden (S), Osten (E), Westen (W), Oben (T), Unten (B) auf den Knoten geführt.
- in einem n-dimensionalen System werden die Busse aus $2 \times n$ Richtungen (je Dimension existiert ein Richtungsvektor \vec{v} dessen Richtung durch das Vorzeichen bestimmt wird \Rightarrow pro Dimension existieren zwei Richtungen, die durch das Vorzeichen des Richtungsvektors unterschieden werden) auf den Knoten geführt.

Innerhalb des Knotens befindet sich ein Bussystem, auf das die externen Busse aufgeschaltet werden und das somit aus mehreren Sammelschienen besteht. Ein externer erster Bus wird über ein Tor auf eine Sammelschiene aufgeschaltet. Über ein weiteres Tor wird die Sammelschiene auf den externen zweiten Bus aufgeschaltet. Um Broadcasting, das Versenden von Daten an mehrere Empfänger, zu ermöglichen, können auch mehrere "zweite" Busse auf das interne Bussystem geschaltet werden. Die Tore können dabei als rein passive Schalter oder als Bustreiber ausgestaltet sein.

Weiterhin hat der Knoten Zugriff auf eine (oder mehrere) mit ihm verbundene(n) konfigurierbaren Elemente (Zellen). Diese verbindet er gegebenenfalls mit einem oder mehreren der angrenzenden Bussegmente.

3.3 Das Routingverfahren

Eine Verbindung wird initiiert von einem Datensender (DS) – dies ist ein konfigurierbares Element (Logikzelle, Buszelle (auch Externe gemäß PACT03)) –, der eine Verbindung zu einem Datenempfänger (DE) – der ebenfalls aus einem konfigurierbaren Element besteht – benötigt. Hierzu teilt der DS dem Knoten seine Busanforderung mit. Der erste Knoten direkt nach einem Datensender wird Initiator-Knoten genannt. Der entnimmt, wie nachfolgend beschrieben, einer internen Tabelle die Adresse des nächsten für den Datentransfers benötigten Knotens. Sofern der Knoten in der Lage ist, das erste benötigte Bussegment anzusprechen – das ist immer dann möglich, wenn das Bussegment und eine interne Sammelschiene des Knoten frei ist – baut er die Verbindung auf, ansonsten weist er die Anforderung des DS zurück und der DS versucht den Zugriff zu einem späteren Zeitpunkt noch einmal, oder wartet und hält die Zugriffsanforderung weiter aufrecht, bis der Knoten sie akzeptiert.

Jeder nachfolgende Knoten entnimmt wiederum die Adresse des nächsten Knotens aus seiner internen Tabelle und setzt den Aufbau der Verbindung fort. Ist der Knoten nicht in der Lage den Aufbau der Verbindung durchzuführen (entweder ist das benötigte Bussegment besetzt oder die Sammelschiene des Knotens ist nicht frei), so kann er entweder in einen Wartezustand springen, oder den Aufbau unterbrechen und einen Fehler an den Initiator-Knoten zurückmelden.

Erst wenn eine Verbindung komplett aufgebaut ist, werden die Datenpakete übertragen und der Sender erhält erhält die Quittierungssignale für den Datenaustausch (vgl. rACK in PACT02). Dadurch wird der Datentransfer automatisch mit den Datensendern synchronisiert. Schlägt der Verbindungsaufbau fehl und muß er zu einem späteren Zeitpunkt wiederholt werden, erhält der Datensender kein Quittierungssignal, somit gehen keine Daten verloren.

Ist die Verbindung komplett aufgebaut, bleibt sie so lange quasi-permanent (das bedeutet, sie erscheint dem DS und DE wie eine direkte Verbindung) stehen, bis der DS die Verbindung durch eine Meldung an den Initiator-Knoten wieder auflöst. Es ist denkbar ein Timeout-Verfahren zu implementieren, das eine stehende Verbindung nach einer gewissen Zeit unterbricht, ins besondere dann, wenn längere Zeit kein Datentransfer stattgefunden hat, um die Bussegmente für andere Verbindungen freizugeben.

3.4 Die Routingtabellen

Grundbaustein eines Knotens ist eine Routing-Tabelle, ähnlich der in PACT04 beschriebenen Switching-Tabellen. Der mögliche Aufbau einer solchen Tabelle ist wird anhand eines Ausführungsbeispiels beschrieben:

Tor				EALU				Bus		Eintrag Position in Zieltabelle	Laufzeit	Aufbauzeit
1	2	3	4	O1	O2	R1	R2	1	0	n..0	n..0	n..0

Jede Zeile stellt eine gültige Verbindung dar. Dabei können mehrere Verbindungen gleichzeitig aktiv sein, maximal so viel Verbindungen, wie freie interne Sammelschienen und freie externe Bussegmente existieren. Eine nachfolgend beschriebene Überwachungslogik übernimmt die Prüfung, ob eine neu ausgewählte Verbindung aufgebaut werden kann. Jede Zeile wird über ihre binäre Adresse angesprochen und ausgewählt. Weiterhin gibt es Sonderzeilen, die keine binäre Adresse besitzen, sondern über Triggersignale oder Zustandssignalen selektiert werden. Zu diesen Signalen gehören (vgl. PACT02)

- rRDYI, rRDYh
- oACK1, oACK2.

Hierbei werden Busverbindungen zu den Datenempfängern (rRDY) automatisch immer dann aufgebaut, wenn die Datensender-Zelle gültige Ergebnisse besitzt. Bei den Signalen oACK ist der Empfänger-/Senderablauf vertauscht. Der Datenempfänger baut die Verbindung zu seinem Datensender auf, sobald der Datenempfänger seine Operanden abgearbeitet hat und in der Lage ist neue Operanden zu verarbeiten.

In den Spalten Tor werden die für die jeweilige Verbindung aktiven Tore eingetragen, dabei ist es möglich, daß nur ein Tor markiert ist und in den Spalten EALU eine Verbindung zu einem konfigurierbaren Element (einer oder mehrerer Zellen) zum Beispiel zu einem R-REG oder O-REG (vgl. PACT02) gewählt wird.

In der Spalte Bus wird die für diese Verbindung verwendete interne Sammelschiene ausgewählt, dabei ist der Wert binär, so daß in der beschriebenen Tabelle insgesamt 4 interne Sammelschienen benutzt werden können. Die Auswahl der internen Sammelschienen kann auch entfallen, in dem ein Prioritätsdekoder die erste freie Sammelschiene erkennt und diese automatisch zuteilt.

In der Spalte Eintrag Position in Zieltabelle wird die Adresse der Zeile der Tabelle angegeben, die den Knoten steuert, zu dem Verbindung aufgenommen wird. An dieser Adresse stehen die für die aktuelle Verbindung benötigten Routinginformationen im nächsten Knoten.

Die Spalte Laufzeit ist optional. Darin kann die erwartete Signallaufzeit vom DS zum DE angegeben werden. Diese Information kann für Berechnungen des Datendurchsatzes oder zur Generierung eines Timeout verwendet werden.

Die Spalte Aufbauzeit ist optional. Darin kann die maximale Zeit zum Aufbauen der Verbindung zum nächsten Knoten (oder der gesamten Verbindung vom DS zum DE) angegeben werden. Wird diese Zeit überschritten kann über einen Timeout der Verbindungsaufbau abgebrochen werden und die Bussegmente und Knoten werden für andere Verbindungen frei. Zu einem späteren Zeitpunkt versucht der DS erneut die Verbindung herzustellen.

Die Einträge in der Routingtabelle können gemäß bekannten Verfahren (PACT02, PACT04) von einer Ladelogik konfiguriert und umkonfiguriert werden.

3.5 Der Arbitr

Der Routingtabelle vorgeschaltet ist ein Arbitr, die aus der Menge der Anfragen zum Verbindungsaufbau über die Routingtabelle einige Signale auswählt. Dabei ist es sinnvoll den Arbitr aus einem gewöhnlichen Priority-Logik und einem sogenannten Round-Robin-Arbitr nach dem Stand der Technik (er schaltet die höchste Priorität auf einer Zeitscheibe immer dem nächsten Signal zu, das bedeutet, das Signal, das momentan die höchste Priorität hat, wird als nächstes die Niederste besitzen und dann mit jedem Zugriff eine Höhere erhalten) aufzubauen. Dabei kann die Priority-Logik dazu dienen einigen Signalen, wie zum Beispiel rACK, oRDY grundsätzlich eine besonders hohe – oder besonders niedere – Priorität zuzuordnen. Der Round-Robin-Arbitr sorgt dafür, daß ein angeforderter Verbindungsaufbau, der soeben nicht aufbaubar war, die niederste Priorität erhält und somit wartet, bis alle anderen Verbindungsanforderungen entweder aufgebaut oder überprüft wurden.

3.6 Die Zustandsmaschine (Steuerung)

Eine Zustandsmaschine steuert die internen Abläufe im Knoten. Dabei ist die Zustandsmaschine zweigeteilt:

1. Die Steuerung des Knotens
2. Die Steuerung des Bustransfers und der Synchronisation. Dabei kann auf eine Zustandsmaschine ähnlich der in PACT02 beschriebenen zurückgegriffen werden.

Die Zustandsmaschine(n) sind gemäß dem Stand der Technik implementierbar und werden nicht näheres beschrieben.

3.7 Die Überwachungseinheit

Eine Überwachungseinheit ist der Routingtabelle nachgeschaltet. Sie entnimmt die in eine adressierte Zeile eingetra-

genen Daten und überprüft ob der Verbindungsaufbau möglich ist. Dabei werden vor allem folgende Punkte überprüft:

- Steht eine freie interne Sammelschiene zur Verfügung.
- Ist das angeforderte Tor frei.
- Ist das angeforderte externe Bussegment frei.
- Ist die Überprüfung positiv, wird das Signal ACCEPT generiert und an die Zustandsmaschinen, sowie die, den Verbindungsaufbau anfordernde Einheit, gesendet, um einen erfolgreichen Aufbau zu signalisieren.
- Ist die Überprüfung negativ, wird das Signal REJECT generiert und an die Zustandsmaschinen, sowie die, den Verbindungsaufbau anfordernde Einheit, gesendet, um einen fehlgeschlagenen Aufbau zu signalisieren. Auf dieses Signal kann der oben beschriebene Arbitrer reagieren und die Priorität dieser Anforderung auf die Niederste setzen.

3.8 Verbindungsabbau

Eine bestehende Verbindung kann durch verschiedene Kriterien wieder abgebaut werden. Zu den wichtigsten Kriterien zählen

- Timeout: Eine Verbindung wird abgebaut, da über eine längere Zeitspanne kein Datentransfer stattgefunden hat. Ein Timeout kann einfach über einen ladbaren Abwärtszähler implementiert werden. Der Zähler wird bei jedem übertragenen Datum neu mit einem festen Ausgangswert, der die Zeitspanne bis zum Timeout repräsentiert, geladen. Findet kein Datentransfer statt, zählt er mit jedem Bustakt um eins herunter. Kommt er bei null an, ist die maximale Zeitspanne abgelaufen und der Bus wird abgebaut.
- Datenzähler: Ein ladbarer Zähler wird mit der zu übertragenden Anzahl von Daten geladen. Bei jedem Datentransfer zählt der Zähler um eins herab. Kommt der Zähler bei null an, sind alle Daten übertragen und der Bus wird abgebaut.
- Synchronisationssignale (vgl. PACT08): Der Busabbau wird durch die Status- und/oder Synchronisationssignale der Zelle(n) gesteuert, die als Datensender auftreten. Zeigen die Zellen beispielsweise an, daß ihre Datenverarbeitung beendet ist oder daß sie umkonfiguriert werden können, wird der Bus abgebaut, da er nicht mehr benötigt wird. Zu diesen Signalen gehören auch rRDY, oACK.

Der Verbindungsabbau geht dabei so vonstatten, daß vom Initiator-Knoten aus ein Signal zum Abbauen der Verbindung gesendet wird. Jeder weitere Knoten sendet das empfangene Signal an seine(n) Partnerknoten weiter und löst die Verbindung sofort auf.

3.9 Broadcasting

In PACT02 ist ein Bussystem beschrieben, das das Senden eines Datenpaketes an mehrere Empfänger erlaubt und ebenfalls eine Quittierung des Datentransfers ermöglicht. Dasselbe System kann bei dem hier beschriebenen Verfahren verwendet werden. Es ist ohne weiteres möglich, mehrere Tore auf eine Sammelschiene zu schalten. Dabei wird nur eine Zeile in der Routingtabelle verwendet. Die Adresse innerhalb der Zielroutingtabellen Eintrag Position in Zieltabelle muß dabei bei jeder der selektierten Routingtabellen zwangsläufig dieselbe sein.

Um dieses Problem zu umgehen, ist es möglich mehrere Einträge für als Eintrag Position in Zieltabelle zur Verfügung zu stellen. Beispielsweise könnte für jedes Tor ein separater Eintrag Position in Zieltabelle existieren. Damit ist jedem Tor eine Adresse innerhalb der Zielroutingtabelle zugeordnet.

Die Quittierung erfolgt genau wie in PACT02 beschrieben über Signalleitungen, die von Open-Kollektor-Treiber getrieben werden und mit einem Transistor abgeschlossen sind.

3.10 Ladelogik (PLU)

Eine Ladelogik im Sinne von DE 44 16 881 A1 ist an die Routingtabelle angeschlossen. Über diese kann die Routingtabelle – äquivalent zu den Switchingtabellen aus PACT04 – konfiguriert und umkonfiguriert werden. Die Routingtabellen können um eine Spalte erweitert werden, die benutzt wird um Rückmeldungen an die Ladelogik zu senden, wenn die in der betreffenden Zeile angegebene Verbindung auf- oder abgebaut wird. So kann in der Spalte angegeben werden, ob eine Rückmeldung an die PLU erfolgen soll, wenn die Verbindung auf- oder abgebaut wird, sowie welcher Art die Rückmeldung ist. Die Rückmeldung erfolgt gemäß PACT02 Fig. 15, wobei statt des Latches 1504 ein Tor eingesetzt ist, das je nach Einstellung entweder beim Verbindungsaufbau oder -abbau durchschaltet und die Rückmeldung an die Transistoren (1502), die als Open-Kollektor-Treiber geschaltet sind ansteuern.

restliche Tabelle	PLU		
	Meldung bei Aufbau	Meldung bei Abbau	Binärer Wert der Rückmeldung
...	b	b	2 1 0

4 Kurzbeschreibung der Diagramme

- Fig. 1: Zeigt einen zweidimensionalen Baustein mit Zellarray und Knoten.
 Fig. 2: Zeigt einen Ausschnitt aus Fig. 1.
 5 Fig. 3: Zeigt mehrere bestehende Busverbindungen und den Aufbau neuer Verbindungen.
 Fig. 4: Zeigt den Aufbau neuer Busverbindungen.
 Fig. 5: Zeigt den nächsten Schritt des Verbindungsaufbaus.
 Fig. 6: Zeigt die Kollision zweier Busse.
 Fig. 7: Zeigt den schrittweisen Verbindungsabbau nach einer Kollision.
 10 Fig. 8: Zeigt den schrittweisen Verbindungsabbau nach einer Kollision.
 Fig. 9: Zeigt den schrittweisen Verbindungsabbau nach einer Kollision.
 Fig. 10: Zeigt den schrittweisen Verbindungsabbau nach einer Kollision.
 Fig. 11: Zeigt den Neuaufbau der abgebauten Verbindung nach einer gewissen Zeitspanne.
 Fig. 12: Zeigt die Fortsetzung von Fig. 6. wenn der Knoten mehr als eine Sammelschiene besitzt und somit keine Kol-
 15 lision stattfindet.
 Fig. 13: Zeigt die Verbindung der Bussegmente mit den Sammelschienen eines Knotens.
 Fig. 14: Zeigt den Datentransfer eines Verbindungsaufbaus. Die Unterfiguren zeigen den Zustand der Verbindung jeweils im Abständen von einem Buszyklus.
 Fig. 15: Zeigt den Datentransfer eines Verbindungsabbaus. Die Unterfiguren zeigen den Zustand der Verbindung jeweils im Abständen von einem Buszyklus.
 20 Fig. 16: zeigt die Steuereinheit eines Knotens.
 Fig. 17: Zeigt die Steuerung eines Knotens.
 Fig. 18: Zeigt das Broadcasting an mehrere Datenempfänger.

5 Detailbeschreibung der Diagramme

Fig. 1 zeigt einen Baustein der Gattungen FPGA, DPGA, DFP (DE 44 16 881 A1). Der Baustein ist zweidimensional symmetrisch aus konfigurierbaren Zellen (0101) aufgebaut, dabei kann 0101 auch eine Mehrzahl von – auch unterschiedlichen – konfigurierbaren Zellen, die zu einer Gruppe zusammengefaßt und untereinander vernetzt sind, repräsentieren. Zwischen den Zellen befinden sich die Knoten des Bussystemes (0102). Dabei sind mehrere Knoten gekennzeichnet, die im folgenden Ausführungsbeispiel mehrere Verbindungen aufbauen werden. Der Datensender A (0103) wird im folgenden eine Verbindung zum Datenempfänger A (0104) aufbauen, sowie der Datensender B (0106) mit dem Datenempfänger B (0105). Eine ausschnittsweise Vergrößerung (0107) ist in Fig. 2 dargestellt.

Fig. 2 zeigt einen Ausschnitt aus einem Baustein beschriebener Gattung. Die konfigurierbare(n) Zeile(n) aus Fig. 1 (0101) sind in 0201 dargestellt. Ein Bündel (0203) aus einer Vielzahl von Leitungen (die Anzahl ist beliebig und nicht exakt dargestellt) verbindet 0201 mit einem Knoten 0202. Die Knoten (0202) sind über die Bussegmente (0205) miteinander verbunden. Ebenfalls haben die Knoten über die Bussegmente 0206, deren Ausgestaltung dieselbe wie die der Bussegmente 0205 ist, Anschluß an die sich außerhalb der Vergrößerung befindenden Knoten. Über das optionale Leitungsbündel (0204) wird verdeutlicht, daß die konfigurierbare(n) Zelle(n) auch über mehrere unterschiedliche Leitungsbündel an eine Mehrzahl von Knoten (0202) angeschlossen sein kann/können.

Fig. 3 zeigt den Baustein zur Laufzeit. Es existieren mehrere Verbindungen:

- zwischen den Knoten Z (0301)
- zwischen den Knoten Y (0306).

Der Datensender A (0302) versucht eine Verbindung (0303) zum Datenempfänger A (0307) aufzubauen. Die Verbindung wird jedoch zurückgewiesen (REJECT), da sie beim Knoten Z (0308) blockiert ist. Gleichzeitig baut der Datensender B (0304) ein Bussegment (0305) zu seinem Empfänger auf. Dieser Versuch gelingt, da der angesprochene Knoten und das benötigte Bussegment nicht blockiert ist.

Fig. 4 zeigt den nächsten Buszyklus. Die Verbindungen Y und Z sind zwischenzeitlich abgebaut. Der Datensender A (0401) kann nun das Bussegment (0403) aufbauen, da der Knoten (0402) nicht mehr blockiert ist. Gleichzeitig baut der Datensender B (0404) das bereits bestehende Bussegment (0405) über den Knoten (0406) hinaus aus → Das neue Bussegment (0407) wird aufgebaut.

Fig. 5 Der in Fig. 3 begonnene und über Fig. 4 fortgesetzte Busaufbau setzt sich äquivalent zu Fig. 4 fort.

Fig. 6 Der Verbindungsaufbau des Bussegmentes 0602 des Datensenders B schlägt fehl. Der Knoten 0601 ist besetzt und liefert ein REJECT-Signal, das das Fehlschlagen des Verbindungsaufbaus signalisiert und den Abbruch der Verbindung bewirkt, an den Knoten 0603.

Fig. 7 zeigt den kompletten Verbindungsaufbau zwischen dem Datensender A (0701) und dem Datenempfänger A (0702). Die Verbindung des Datensenders B wird weiter abgebaut. Der Knoten 0703 liefert über das Bussegment (0704) das REJECT-Signal an den Knoten 0705. Daraufhin wird das Segment 0704 abgebaut.

Fig. 8 Der Datentransfer zwischen dem Datensender A und dem Datenempfänger A beginnt. Die Verbindung des Datensenders B wird weiter abgebaut. Der Knoten 0801 liefert über das Bussegment (0802) das REJECT-Signal an den Knoten 0803. Daraufhin wird das Segment 0802 abgebaut.

Fig. 9 Der Datentransfer zwischen dem Datensender A und dem Datenempfänger A setzt sich fort. Das letzte Segment des Datensenders B wird abgebaut. Der Knoten 0901 liefert über das Bussegment (0902) das REJECT-Signal an den Knoten 0903. Daraufhin wird das Segment 0902 abgebaut.

Fig. 10: Der Datentransfer zwischen dem Datensender A und dem Datenempfänger A setzt sich fort. Der Datensender B (1001) wartet eine Zeitspanne ab, bis er erneut versucht eine Verbindung zu seinem Datenempfänger aufzubauen.

Fig. 11 zeigt den Zustand mehrere Buszyklen später: Der Datentransfer zwischen dem Datensender A und dem Datenempfänger A findet immer noch statt. Der Datensender B (1101) startet einen neuen Versuch die Verbindung zu seinem Datenempfänger aufzubauen. Er baut ein Bussegment (1102) zum Knoten (1103) auf. Sofern in den nächsten Buszyklen die Verbindung des Datensenders A zu seinem Datenempfänger A abgebaut ist, gelingt der Verbindungsaufbau des Datensenders B (1101), ansonsten schlägt sie wie in Fig. 6 beschrieben erneut fehl.

Fig. 12 zeigt die Fortsetzung der Fig. 6. wenn der Knoten 1202 in der Lage ist, mehr als eine Verbindung aufzubauen, das heißt, der Knoten besitzt mehrere interne Sammelschienen. Dadurch kann die Verbindung des Datensenders A über die erste Sammelschiene abgewickelt werden und die Verbindung des Datensenders B über die zweite. Der Knoten 1202 baut das Bussegment 1203 zu dem Datenempfänger B (1201) auf.

Fig. 13 stellt eine knoteninterne Vernetzungsstruktur dar. Es existieren vier knoteninterne Sammelschienen 1301, 1302, 1303, 1304. Über je eine Gruppe von Toren (1308, 1309, 1310, 1311) werden die Sammelschienen auf die Bussegmente Westen (1318), Norden (1316), Osten (1319), Süden (1317) geschaltet. Über die Tore 1307 werden die Sammelschienen mit den O-REG1/2 (vgl. PACT02) verbunden (1314, 1315). Über die Tore 1306 wird das R-REG (vgl. PACT02) auf die Sammelschienen geschaltet. Dabei kann das niederwertige und höherwertige R-REG (1312, 1313) jeweils getrennt verschaltet werden. Die Tore werden über den Bus 1320 angesteuert. Dabei wird das benötigte Torsystem (Westen, Norden, Osten, Süden), sowie die interne Sammelschiene angegeben. Durch Verbindung (1325, 1326, 1327, 1328) der Richtungsangabe mit der Angabe der selektierten Sammelschiene wird das erforderliche Tor (1321, 1322, 1323, 1324) ausgewählt.

Fig. 14a zeigt den Aufbau einer Verbindung, dabei übermittelt der Datensender (1401) dem Knoten des Datensenders (1402) das erste Datenpaket.

Fig. 14b Der Knoten selektiert dabei aus der Routingtabelle den zu rRDY – rRDY ist ein Statussignal, das anzeigt, daß Daten am Datensender bereitstehen (vgl. PACT02) – gehörenden Eintrag. Anhand diesem Eintrag wird das nächste Bussegment aufgebaut und die Adresse der Routingtabelle des nächsten Knotens übertragen.

Fig. 14c Der letzte Knoten (1403), der Datenempfängerknoten, erhält die Adresse für den Eintrag innerhalb seiner Routingtabelle. Der Eintrag zeigt auf keinen weiteren Knoten, sondern auf eine Zelle. Daher aktiviert der Knoten sofort seine Tore zur selektierten Zelle.

Fig. 14d Die Daten gelangen durch das aktivierte Tor von 1403 direkt zur Empfängerzelle 1404.

Fig. 14e Die Zelle sendet das oACK-Signal (vgl. PACT02) als Quittierung für die erhaltenen Daten zurück. Im nächsten Buszyklus (vgl. Fig. 14e) wird der Datensender daraufhin das nächste Datenpaket absenden.

Fig. 14e-g Der normale Datenaustausch zwischen den Zellen läuft ab.

Fig. 15a Es existiert eine Datenverbindung vom Datensender (1501) zum Datenempfänger (1503) über mehrere Knoten.

Fig. 15b Der Datensender (1501) hat seine Datenübertragung beendet und sendet ein DISCONNECT-Signal zum ersten Knoten.

Fig. 15c Das erste Bussegment ist abgebaut und der Knoten reicht das DISCONNECT-Signal weiter.

Fig. 15d Der Verbindungsabbau geht weiter.

Fig. 15d Der letzte Knoten erhält das DISCONNECT. Daraufhin baut der letzte Knoten gleichzeitig die Verbindung zum Vorgängerknoten und zum Datenempfänger ab.

Fig. 15e Das letzte Bussegment und die Verbindung zum Datenempfänger ist abgebaut.

Fig. 16 Zeigt die Steuerung eines Knotens. Die Anforderungen (1601) zum Aufbau einer Verbindung gelangen über den Prioritätsdekoder (1602) an die Routingtabelle (1603). Der Prioritätsdekoder wählt die Anforderung mit der höchsten Priorität aus, wobei eine soeben fehlgeschlagene Anforderung die niederste Priorität zugewiesen bekommt. Die Prioritätslogik erhält ihre Anforderungen über Statussignale (z. B. der der konfigurierbaren Zellen (vgl. PACT02, rRDY, oACK) oder über die Bussegmente 1316, 1317, 1318, 1319. Liegen Daten an den Bussegmenten an, ohne daß das Tor des Bussegmentes aktiviert ist, werden die Daten von der Prioritätslogik als Adresse der Routingtabelle interpretiert und als Anforderung betrachtet. Liegen Statussignale an der Prioritätslogik an (rRDY, oACK), werden diese in Adressen für die Routingtabelle übertragen. Die Adressen der Routingtabelle selektieren einen Eintrag. Die Daten des Eintrages (1604) werden an eine Einheit aus UND-Gattern (1605) weitergeleitet. Dabei wird die Binärzahl der Busauswahl (BUS 1...0) über einen 2 : 4 Dekoder (1606) in Selektsignale übertragen. Die UND-Gatter-Einheit verUNDet dabei jedes Signal mit demselben in einem Latch (1607) gespeicherten Signal. Das heißt daß das Signal GATE1 der Routingtabelle mit dem Signal GATE1 des Latches verUNDet wird, das Signal GATE2 der Routingtabelle mit dem Signal GATE2 des Latches und so weiter. Dabei repräsentieren die Signale des Latches den Zustand der momentan aktuellen Vernetzungsstruktur, d. h. die benutzten Tore und die benutzten Sammelschienen sind in das Latch eingetragen. Ergibt eine VerUNDUNG der Verbindungsanforderung mit dem aktuellen Zustand einen wahren Pegel, bedeutet dies, daß die neue Verbindungsanforderung Ressourcen benötigt, die momentan benutzt sind. Über ein ODER-Gatter (1608) werden alle UND-Gatter miteinander verknüpft. Ergibt sich am Ausgang des ODER-Gatters ein wahrer Pegel, wird die Verbindungsanforderung zurückgewiesen (REJECT) (1609), da die benötigten Ressourcen belegt sind. Aus dem REJECT-Signal wird über einen Inverter (1610) das ACCEPT-Signal (1611) generiert. Die Signale werden an eine Zustandsmaschine (1612), die nach dem Stand der Technik implementiert sein kann, weitergeleitet. Diese steuert danach die Annahme der Verbindung oder die Zurückweisung. Wird die Verbindungsanforderung zurückgewiesen, meldet (1613) die Zustandsmaschine den REJECT an den Prioritätsdekoder und die Anforderung erhält niederste Priorität. Bei einer Annahme der Anforderung, werden die neuen Zustandssignale mit den aktuellen Zustandssignalen nach dem Latch verODERT (1614) – dabei ist die ODER-Einheit ebenso aufgebaut wie die beschriebene UND-Einheit (1605) – und in das Latch (1607) zurückgeschrieben.

Über 1623 steuert die Zustandsmaschine, ob die ODER-Einheit (1614) oder die Maske (1616) aktiv ist. Das Latch wird über 1622 von der Zustandsmaschine getriggert. Über den Bus 1615 gelangt die neue Einstellung an die Tore.

Der Abbau einer Busverbindung läuft auf ähnliche Weise. Allerdings findet muß bei einer Überprüfung der Ressourcen das Signal REJECT auftreten, da die Busverbindung, die abgebaut werden soll, existieren muß. Aufgrund des RE-

JECTs aktiviert die Zustandsmaschine 1612 anstatt der ODER-Einheit 1614 die Maske 1616. Die Verbindungsdaten der abzubauenen Verbindung werden aus dem aktuellen Vernetzungszustand ausmaskiert und in das Latch 1607 zurückgeschrieben. Vor dem Zurückschreiben der neuen Verbindungsdaten sendet die Zustandsmaschine das DISCONNECT-Signal zum Abbau der Verbindung an den nächsten Knoten.

- 5 Über die Tore 1617, 1618, 1619, 1620 kann die Steuerung direkt auf die Sammelschienen 1301, 1302, 1303, 1304 zugreifen. Damit kann die Zustandsmaschine sowohl Steuersignale (DISCONNECT) auf die Sammelschiene übertragen, als auch Steuersignale von der Sammelschiene erhalten (REJECT, ACCEPT) und auf diese reagieren. Ebenfalls dienen diese Tore um den Eintrag Position in Zieltabelle (über 1621) auf die Sammelschiene zu übertragen.

Über 1624 kann die Ladelogik (PLU) auf die Routingtabelle zugreifen.

- 10 Fig. 17: Zeigt ein Broadcasting eines Datensenders (1701) an mehrere Datenempfänger (1702) über eine Vielzahl von Knoten (1707), auf die nicht näher eingegangen wird. Der Bus ist zur Verdeutlichung aufgetrennt dargestellt, nämlich in die Quittierungsleitung (ACK) (1703) und den restlichen Bus (1704). ACK wird dabei negiert und an den wiederum invertierenden Open-Kollektor-Bustreiber geliefert. ACK wird über einen PullUp-Widerstand 1705 auf H gezogen. Dabei ist die Schaltung so aufgebaut, daß folgende Fälle eintreten:

- 15 1. Wird der entsprechende Bus nicht angesteuert, liegt an der Basis des Transistors (1706) L an. Dadurch belastet er den Bus nicht.
 2. Wird der entsprechende Bus angesteuert, und das Signal nicht quittiert, liegt an der Basis des Transistors (1706) H an. Das bedeutet, daß der Bus auf L gezogen wird. Wird ein Ergebnis per Broadcasting an mehrere Datenempfänger verteilt, so ziehen alle Knoten, die die Ergebnisdaten noch nicht quittiert haben und Warte-Zyklen benötigen den Bus auf L.
 20 3. Wird der entsprechende Bus angesteuert, und das Signal quittiert, liegt an der Basis des Transistors (1706) L an. Das bedeutet, daß der Bus nicht belastet wird. Wird ein Ergebnis per Broadcasting an mehrere Datenempfänger verteilt, so belasten alle Knoten, die die Ergebnisdaten quittiert haben und keine Warte-Zyklen benötigen, den Bus nicht.
 25

- Da der Bus in seinem Grundzustand den H-Pegel, also die Quittierung, einnimmt, übersteuert die Nichtquittierung gemäß Fall 2 die Quittierung, indem sie den Bus auf L zieht. Dabei geht der Bus erst dann in den H-Pegel, also in den Quittierungszustand, wenn alle an der Verbindung beteiligten Knoten quittieren. Es ist somit eine Wired-AND-Schaltung realisiert.
 30

6 Begriffsdefinition

- ALU Arithmetisch logische Einheit. Grundeinheit zum Verarbeiten von Daten. Die Einheit kann arithmetische Operationen wie Addition, Subtraktion, unter Umständen auch Multiplikation, Division, Reihenentwicklungen usw. durchführen. Dabei kann die Einheit als ganzzahlige (integer) Einheit oder als Fließkomma-(floating-point)-Einheit gestaltet sein. Ebenfalls kann die Einheit logische Operationen, wie UND, ODER, sowie Vergleiche durchführen.

Arbiter Einheit um Rechte zwischen Signalen zu verteilen.

Broadcast Senden von Daten einer PAE an mehrere oder alle Datenempfänger.

- 40 Busanforderung Die Anforderung eine Busverbindung zum Datentransfer aufzubauen. (Auch Verbindungsanforderung).
 Bussegment Abschnitt eines Bussystems zwischen zwei Knoten.

Datenempfänger Die Einheit(en), die Ergebnisse der PAE weiterverarbeitet/arbeiten.

Datensender Die Einheit(en), die Daten für die PAE als Operanden zur Verfügung stellt/stellen.

Datentyp Art der Daten: Zeichen, Zahlen, Fließkommazahlen, Signale (boolean), etc.

- 45 DFP Datenflußprozessor nach Patent/Offenlegung DE 44 16 881.

DPGA Dynamisch konfigurierbare FPGAs. Stand der Technik.

EALU Erweiterte arithmetisch logische Einheit. ALU, die um Sonderfunktionen, die zum Betrieb einer Datenverarbeitungseinrichtung gemäß DE 44 16 881 A1 benötigt werden oder sinnvoll sind erweitert wurde. Dies sind insbesondere Zähler.

- 50 Elemente Sammelbegriff für alle Arten von in sich abgeschlossenen Einheiten, welche als Stück in einem elektronischen Baustein zum Einsatz kommen können. Elemente sind also:

- Konfigurierbare Zellen aller Art
- Cluster
- 55 - RAM-Blöcke
- Logik
- Rechenwerke
- Register
- Multiplexer
- 60 - I/O Pins eines Chips

FPGA Programmierbarer Logikbaustein. Stand der Technik.

Gatter Gruppe von Transistoren, die eine logische Grundfunktion durchführen. Grundfunktionen sind z. B. NAND, NOR, Transmission-Gates.

- 65 H-Pegel Logisch 1 Pegel, abhängig von der verwendeten Technologie.

Knoten Element das mehrere Bussegmente miteinander verbindet und den Verbindungsaufbau aktiv steuert, während des Datentransfers je doch passiv ist.

konfigurierbares Element Ein konfigurierbares Element stellt eine Einheit eines Logik-Bausteines dar, welche durch ein

Konfigurationswort für eine spezielle Funktion eingestellt werden kann. Konfigurierbare Elemente sind somit, alle Arten von RAM-Zellen, Multiplexer, Arithmetische logische Einheiten, Register und alle Arten von interner und externer Vernetzungsbeschreibung etc.
 konfigurierbare Zelle Siehe Logikzellen.
 Konfigurieren Einstellen der Funktion und Vernetzung einer logischen Einheit, einer (FPGA)-Zelle oder einer PAE (vgl. 5 umkonfigurieren).
 Ladelogik Einheit zum Konfigurieren und Umkonfigurieren der PAE. Ausgestaltet durch einen speziell an seine Aufgabe angepaßten Mikrokontroller.
 Latch Speicherelement, das ein Signal für gewöhnlich während des H-Pegels transparent weiterleitet und während des L-Pegels speichert. 10
 In PAEs werden teilweise Latches gebraucht, bei denen die Funktion der Pegel genau umgekehrt ist. Hierbei wird vor den Takt eines üblichen Latch ein Inverter geschaltet.
 Logikzellen Bei DFPs, FPGAs, DPGAs verwendete konfigurierbare Zellen, die einfache logische oder arithmetische Aufgaben gemäß ihrer Konfiguration erfüllen.
 LookUp-Tabelle Tabelle, die einen Wert als Adresse erhält und ein Ergebnis zurückliefert. Beispielsweise wird eine Zahl 15 als Adresse angegeben und deren sinus wird zurückgegeben.
 L-Pegel Logisch 0 Pegel, abhängig von der verwendeten Technologie.
 M-PLUREG Register in dem die Vernetzung der PAE gesetzt wird. Das Register wird von der PLU beschrieben.
 Open-Kollektor Schaltungstechnik, bei der der Kollektor eines Transistors an einem, über einen Pullup auf den H-Pegel gezogenen, Bussignal liegt. Der Emittor der Transistors liegt auf Masse. Schaltet der Transistor, so wird das Bussignal auf den L-Pegel gezogen. Vorteil des Verfahrens ist, daß eine Mehrzahl solcher Transistoren den Bus ohne elektrische Kollision steuern können. Dabei sind die Signale ODER-verküpft, es entsteht das sog. wired-OR. 20
 PAE Processing Array Element: EALU mit O-REG, R-REG, R2OMUX, F-PLUREG, M-PLUREG, BM-, SM-, Sync-, StateBack- und Power-UNIT.
 Partnerknoten Knoten mit dem ein bestimmter Knoten über ein Bussegment Kontakt hat oder aufbauen will. 25
 PLU Einheit zum Konfigurieren und Umkonfigurieren der PAE. Ausgestaltet durch einen speziell an seine Aufgabe angepaßten Mikrokontroller.
 Prioritätsdekoder Das Signal mit der höchsten Priorität wird weitergegeben oder freigeschaltet.
 Priority-Logik Das Signal mit der höchsten Priorität wird weitergegeben oder freigeschaltet.
 PullDown Widerstand, der eine Busleitung auf einen L-Pegel zieht. 30
 PullUp Widerstand, der eine Busleitung auf einen H-Pegel zieht.
 Routingtabelle Tabelle innerhalb eines Knotens, die Informationen über aufzubauende Verbindungen enthält.
 Round-Robin-Arbiter Arbiter, der der Reihe nach ein Signal nach dem anderen freischaltet. Das aktuell freigeschaltete Signal bekommt die niederste Priorität und wird als Letztes in der Kette dann wieder freigeschaltet. Der Arbiter arbeitet im Kreis. 35
 Sammelschiene Bus auf dem mehrere Bussegmente zusammengeführt werden.
 Switching-Tabelle Eine Switching-Tabelle ist ein Ring-Speicher, welcher durch eine Steuerung angesprochen wird. Die Einträge einer Switching-Tabelle können beliebige Konfigurationswörter aufnehmen. Die Steuerung kann Befehle durchführen. Die Switching-Tabelle reagiert auf Triggersignale und konfiguriert konfigurierbare Elemente anhand eines Eintrages in einem Ringspeicher um. 40
 Timeout Nach Ablauf einer bestimmten Zeit geschieht etwas (ein Vorgang wird begonnen oder unterbrochen).
 Tor Schalter, der ein Signal weiterleitet oder sperrt. Einfacher Vergleich: Relais.
 Umkonfigurieren Neues Konfigurieren von einer beliebigen Menge von PAEs während eine beliebige Restmenge von PAEs ihre eigenen Funktionen fortsetzen (vgl. konfigurieren).
 Verbindungsanforderung Die Anforderung eine Busverbindung zum Datentransfer aufzubauen (auch Busanforderung). 45
 Zellen Synonym für konfigurierbare Elemente.
 Zustandsmaschine Logik, die diversen Zuständen annehmen kann. Die Übergänge zwischen den Zuständen sind von verschiedenen Eingangsparametern abhängig. Diese Maschinen werden zur Steuerung komplexer Funktionen eingesetzt und entsprechen dem Stand der Technik. 50

7 Konventionen

7.1 Namenskonvention

Baugruppe -UNIT 55
 Betriebsart -MODE
 Multiplexer -MUX
 Negiertes Signal not-
 Register für PLU sichtbar -PLUREG
 Register intern -REG 60
 Schieberegisters -sft

7.2 Funktionskonvention

UND-Funktion & 65

5

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

ODER-Funktion #

10

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

15

NICHT-Funktion!

20

A	Q
0	1
1	0

25 TOR-Funktion G

30

EN	D	Q
0	0	-
0	1	-
1	0	0
1	1	1

35

Patentansprüche

1. Bussystem für DFPs gemäß DE 44 16 881 A1, sowie Bausteine mit zwei- oder mehrdimensionaler programmierbarer Zellstruktur (der Gattungen FPGA, DPGA, o. dgl.), **dadurch gekennzeichnet**, daß
1. das Bussystem aus einer Vielzahl voneinander elektrisch unabhängiger Bussegmente besteht, die über Knoten getrennt sind,
 2. die Knoten Bussegmente aktiv zusammenfügen oder trennen, wobei zum Zusammenfügen mehrere Bussegmente über Tore auf eine, sich innerhalb des Knoten befindende Sammelschiene zusammengeschaltet werden,
 3. jeder Knoten eine Routingtabelle besitzt, in der die Informationen über den Aufbau der Verbindungen gespeichert sind,
 4. jeder Knoten eine Überwachungseinheit besitzt, die selbständig prüft ob eine Verbindung aufgebaut werden kann oder nicht.
2. Bussystem nach Patentanspruch 1, dadurch gekennzeichnet, daß eine Busverbindung schrittweise aufgebaut wird, wobei der Datensender zunächst nur die Verbindung zu dem mit ihm verbundenen Knoten herstellt, der Knoten dann über seine Routingtabelle feststellt, welcher ihm benachbarte Knoten für die Verbindung benötigt wird und zu diesem über das verbindende Bussegment Verbindung aufnimmt, sofern dieses nicht bereits von einer anderen Verbindung belegt ist; ist dies der Fall, so wird der Aufbau der Busverbindung abgebrochen; ist dies nicht der Fall, so setzt der angesprochene Knoten den Aufbau der Busverbindung fort.
3. Bussystem nach Patentanspruch 1 und 2, dadurch gekennzeichnet, daß bei einem fehlgeschlagenen Verbindungsaufbau der Aufbau zu einem späteren Zeitpunkt erneut versucht wird.
4. Bussystem nach Patentanspruch 1, 2 und 3, dadurch gekennzeichnet, daß nach einem erfolgten Verbindungsaufbau die Knoten passiv sind und der Datentransfer ausschließlich von Datensender und Datenempfänger gesteuert werden.
5. Bussystem nach Patentanspruch 1, 2, 3 und 4, dadurch gekennzeichnet, daß weder Datensender noch Datenempfänger, weder während des Verbindungsaufbaus noch des Betriebs, Kenntnis über das Bussystem haben müssen und keinerlei aktiven Eingriff auf die Knoten nehmen müssen.
6. Bussystem nach Patentanspruch 1, 2, 3, 4 und 5, dadurch gekennzeichnet, daß die Knoten und deren Routingtabellen von einer Ladelogik konfiguriert und umkonfiguriert werden.
7. Bussystem nach Patentanspruch 1, 2, 3, 4, 5 und 6, dadurch gekennzeichnet, daß über die Routingtabellen Rückmeldungen an die Ladelogik gesendet werden können.
8. Bussystem nach Patentanspruch 1, 2, 3, 4, 5, 6 und 7, dadurch gekennzeichnet, daß das ein Knoten Daten an

DE 197 04 742 A 1

mehrere andere Knoten gleichzeitig versenden kann (Broadcasting).

Hierzu 17 Seite(n) Zeichnungen

5

10

15

20

25

30

35

40

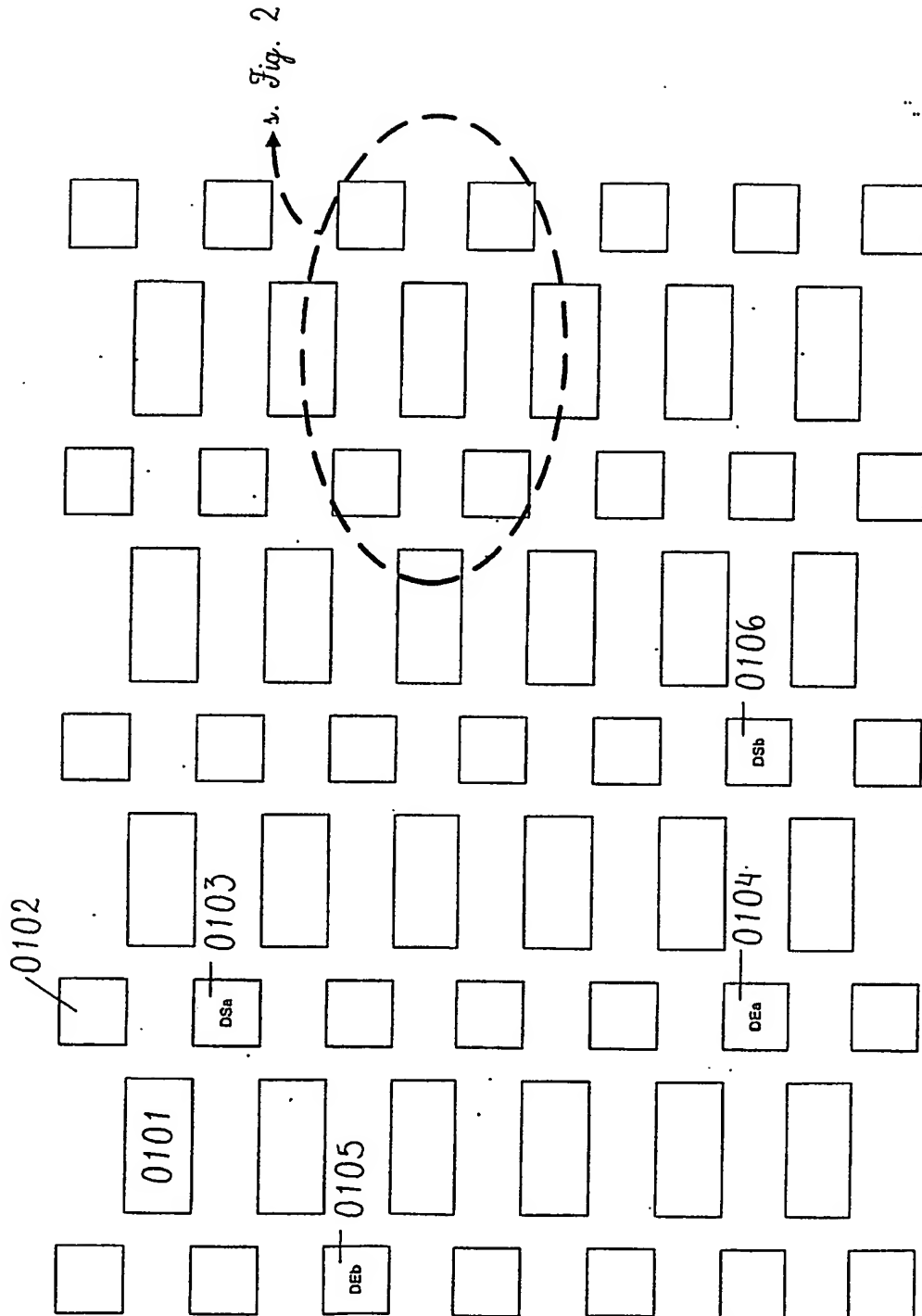
45

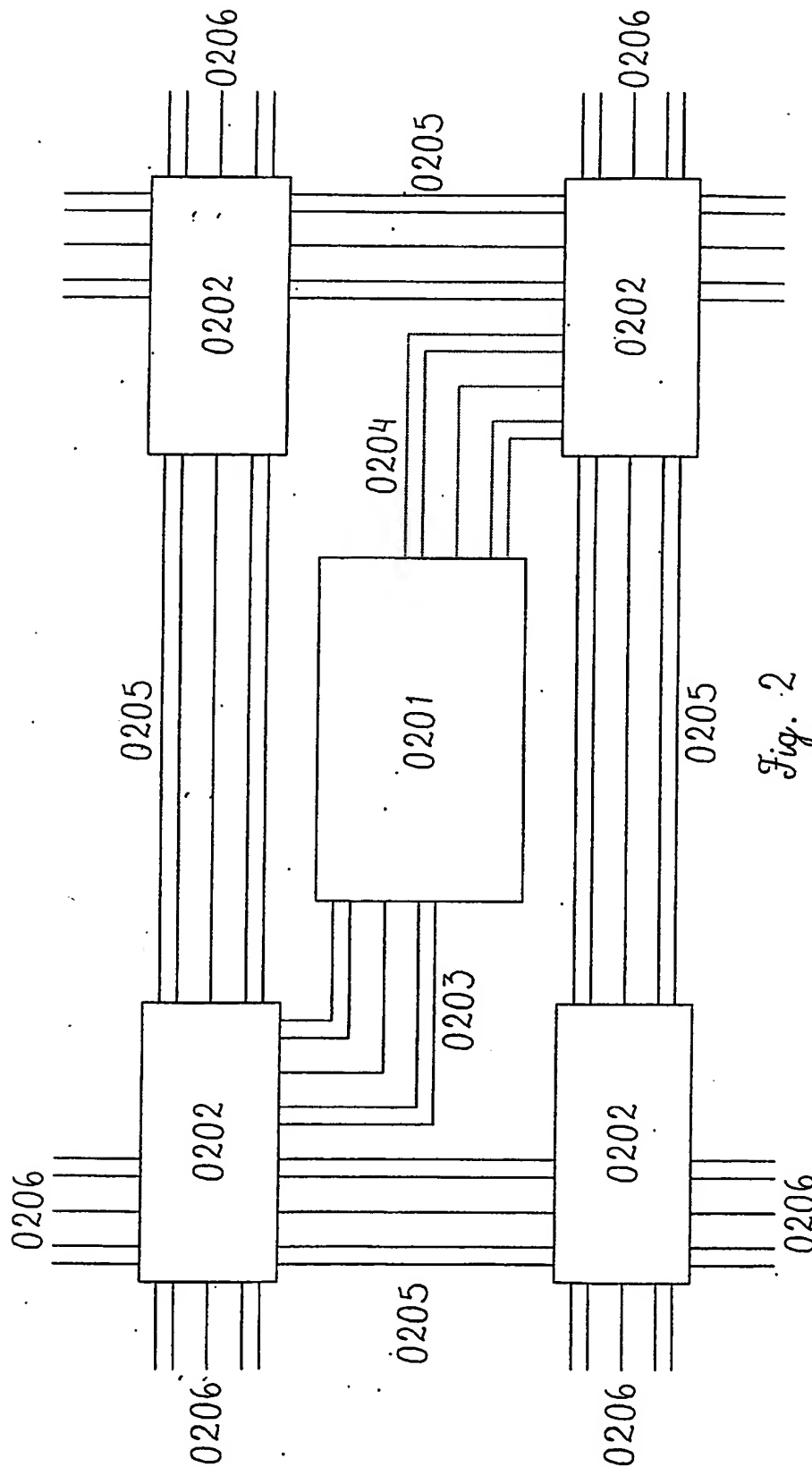
50

55

60

65





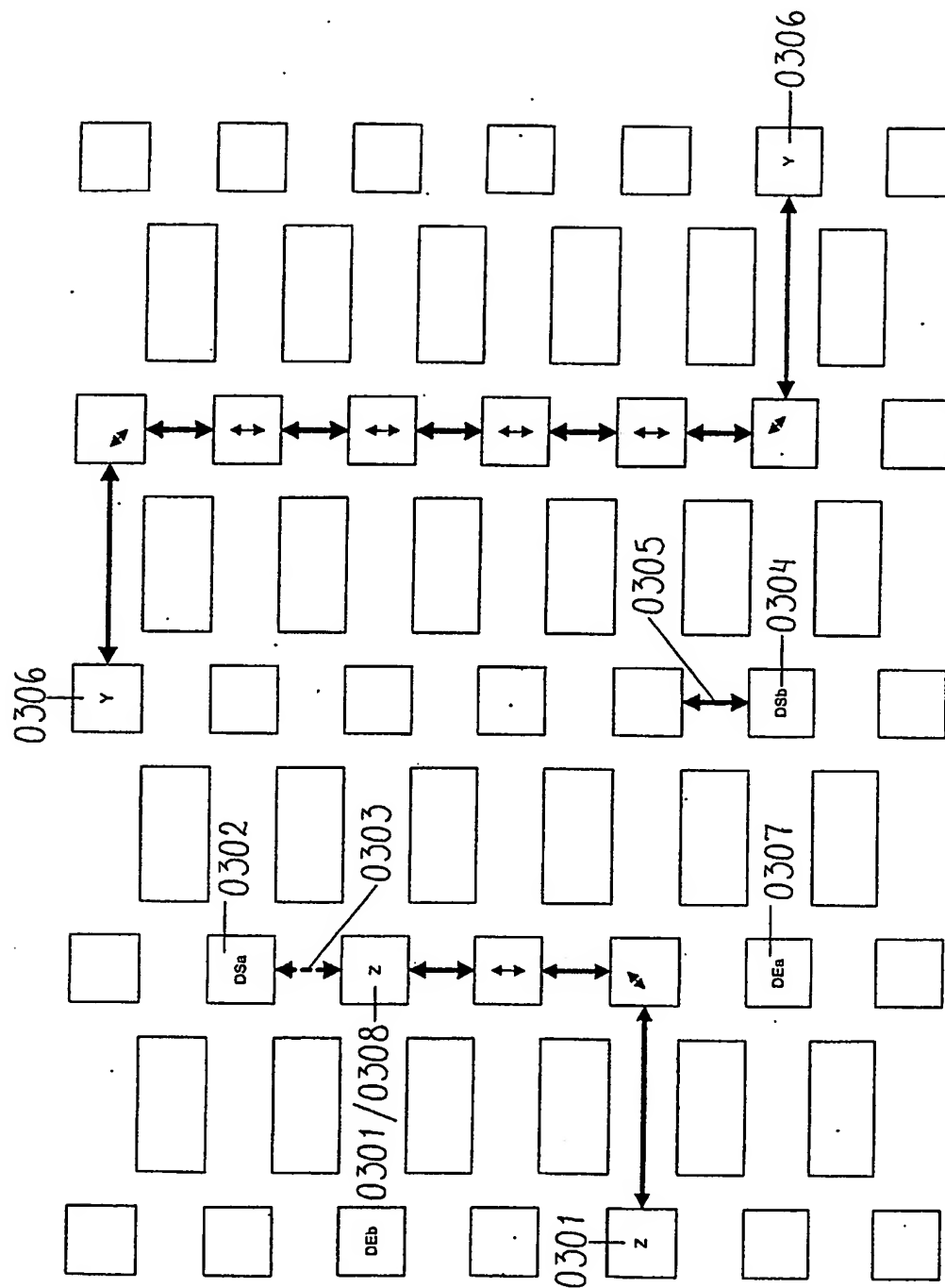
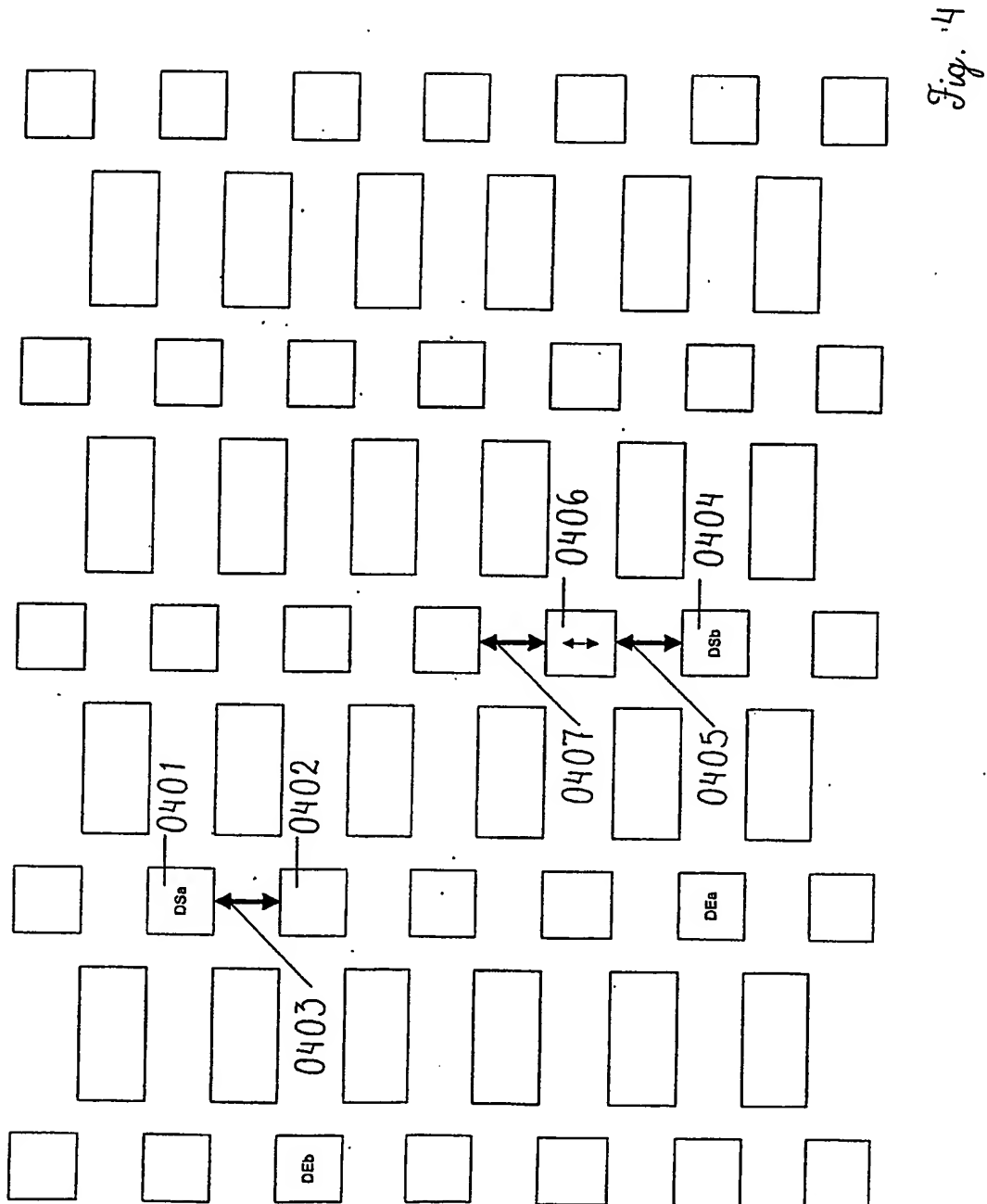
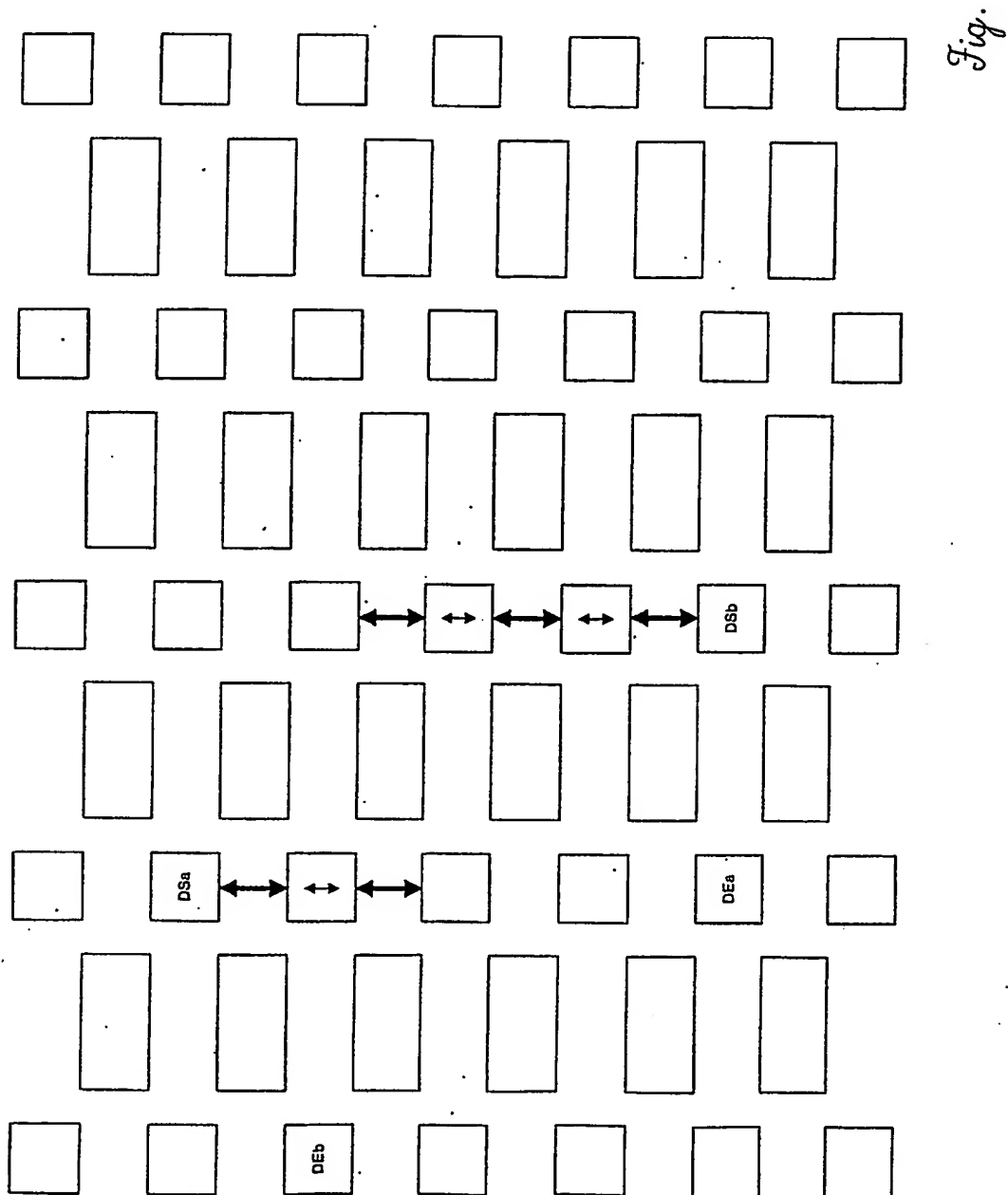
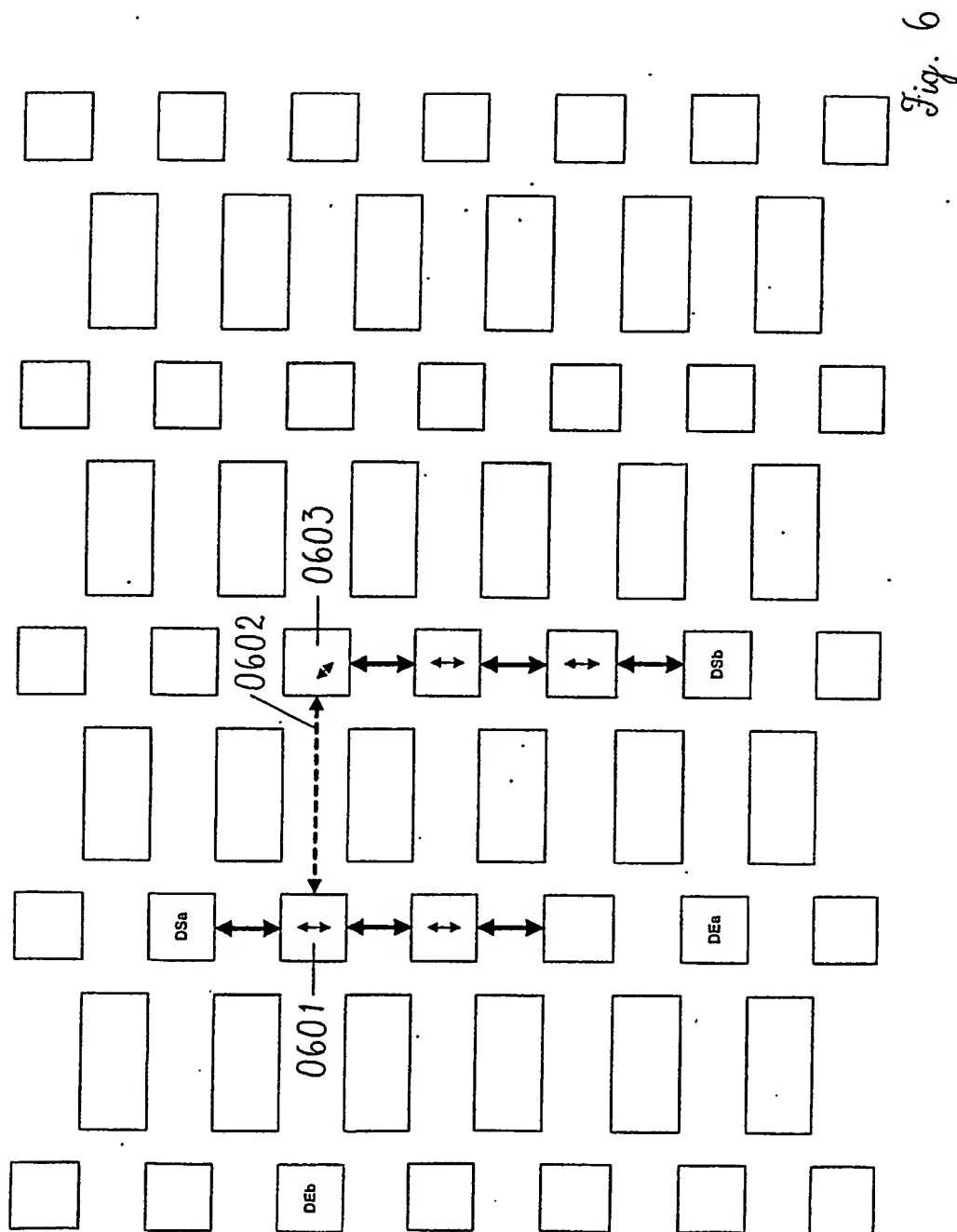
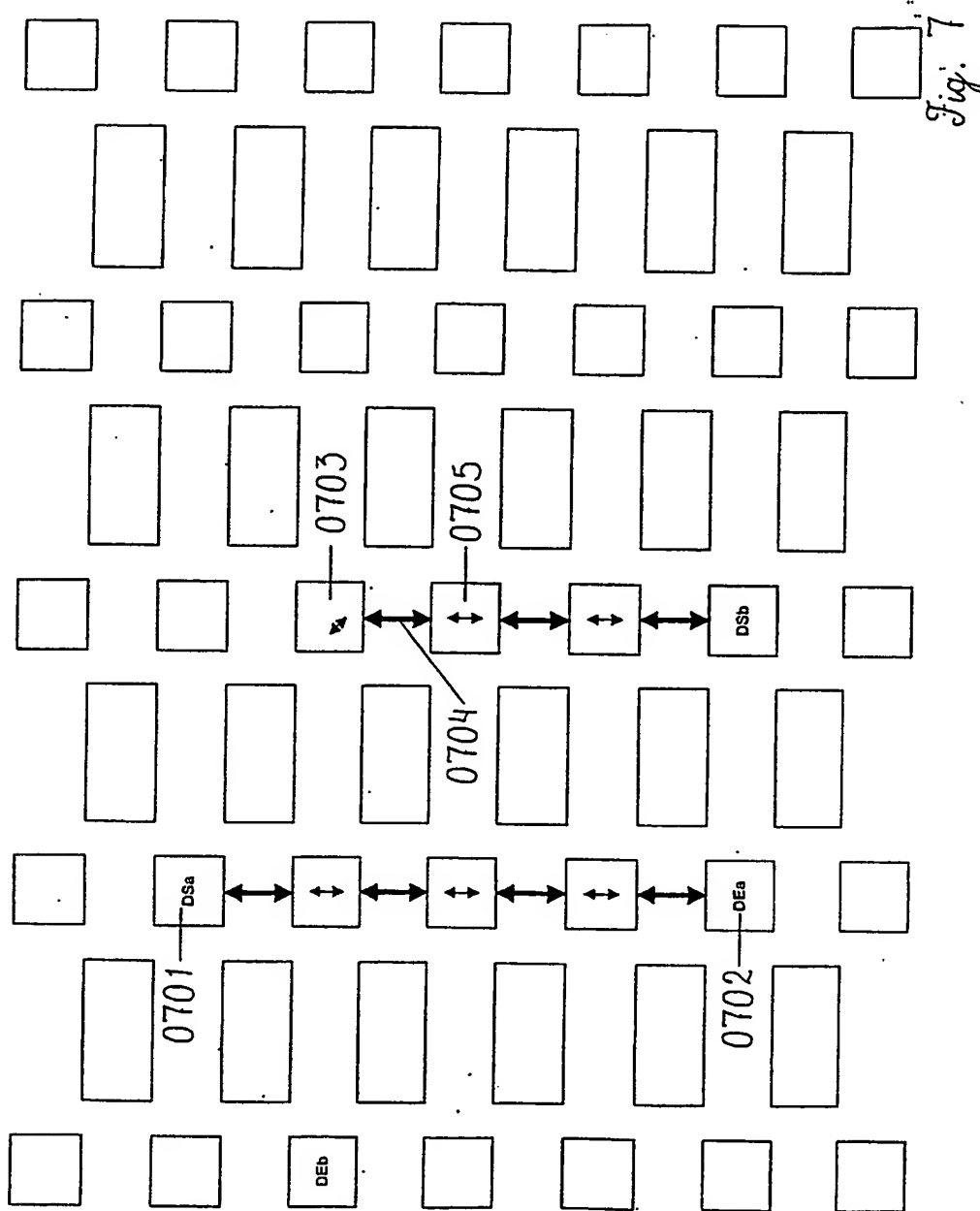


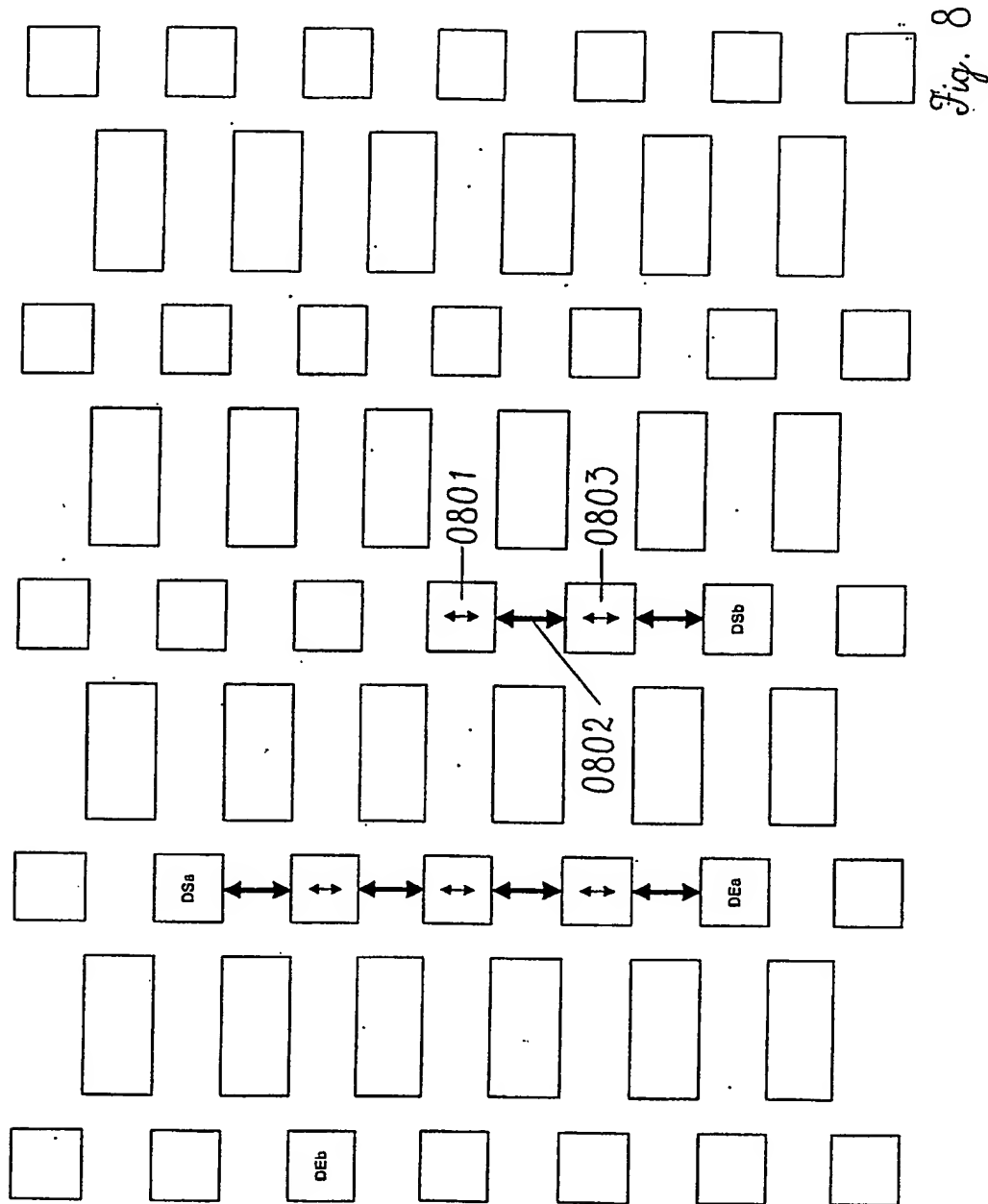
Fig. 3

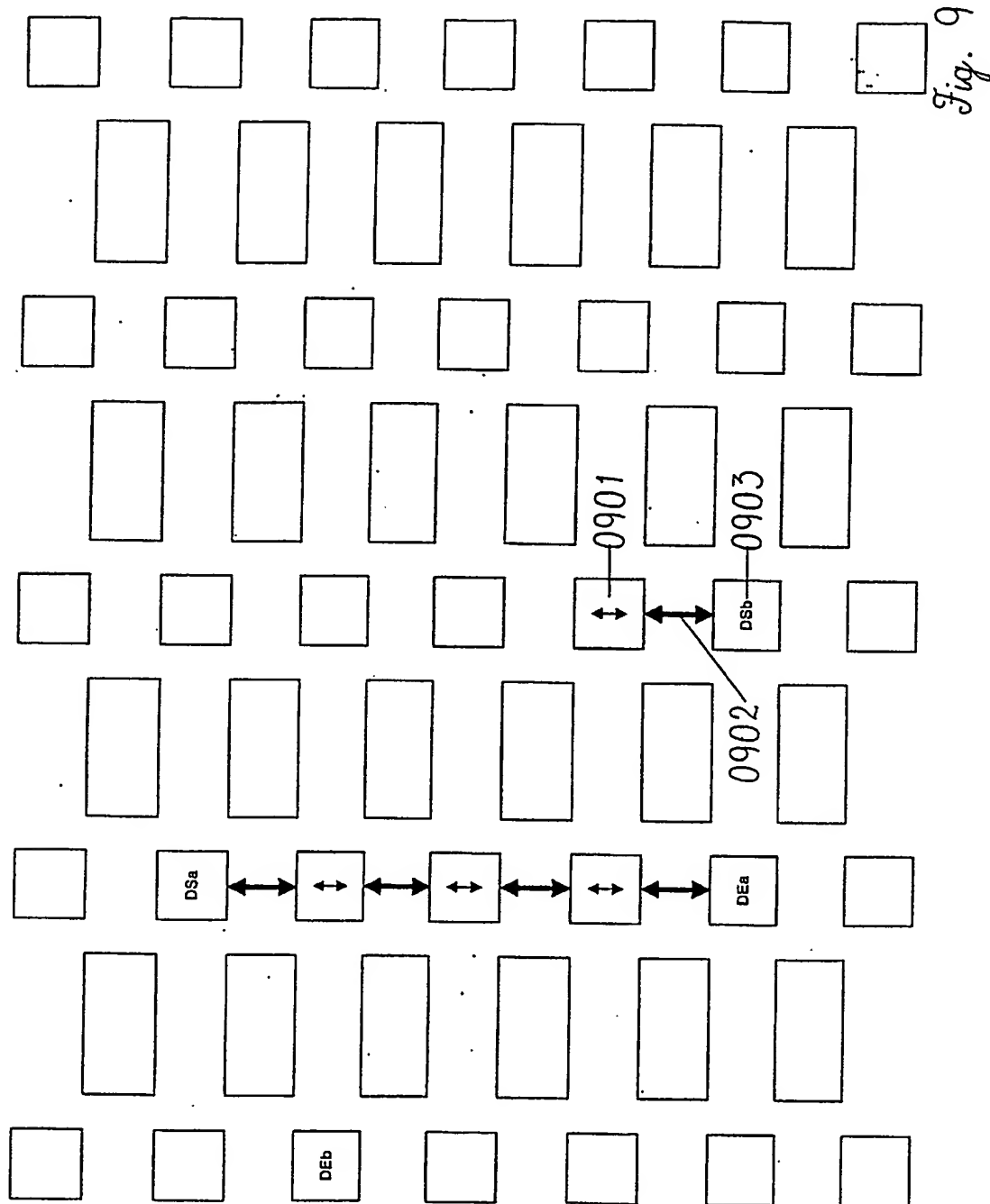












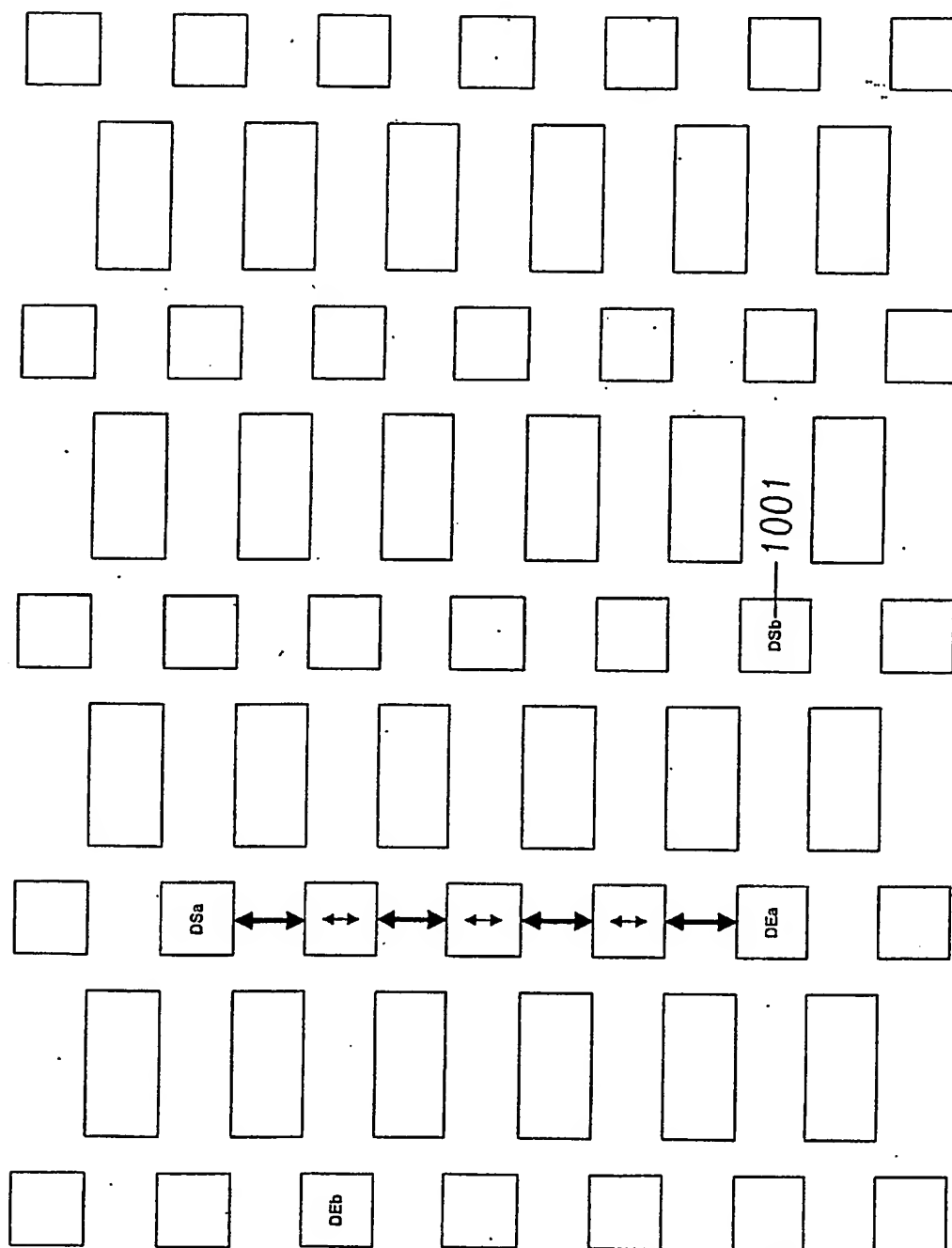
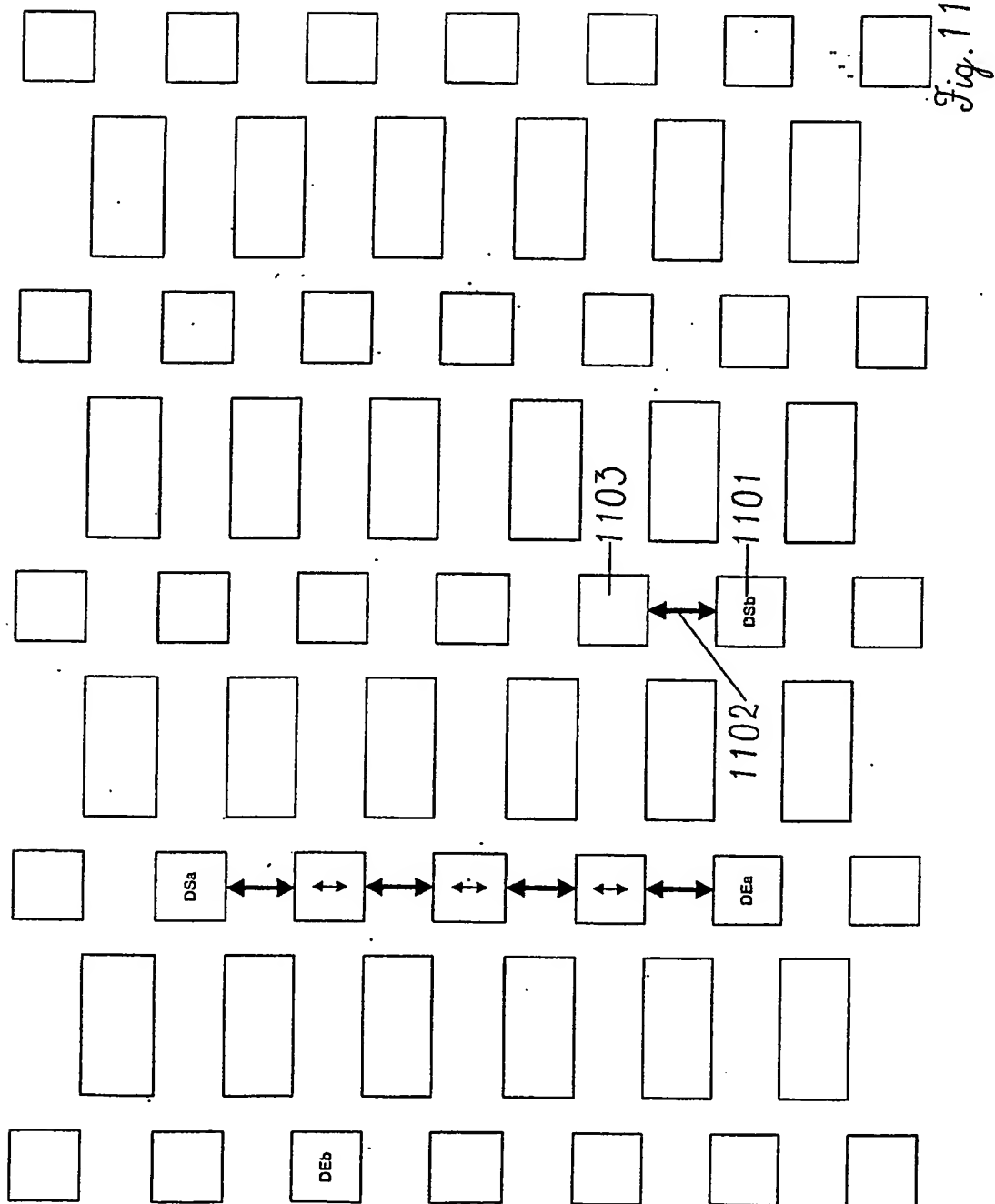
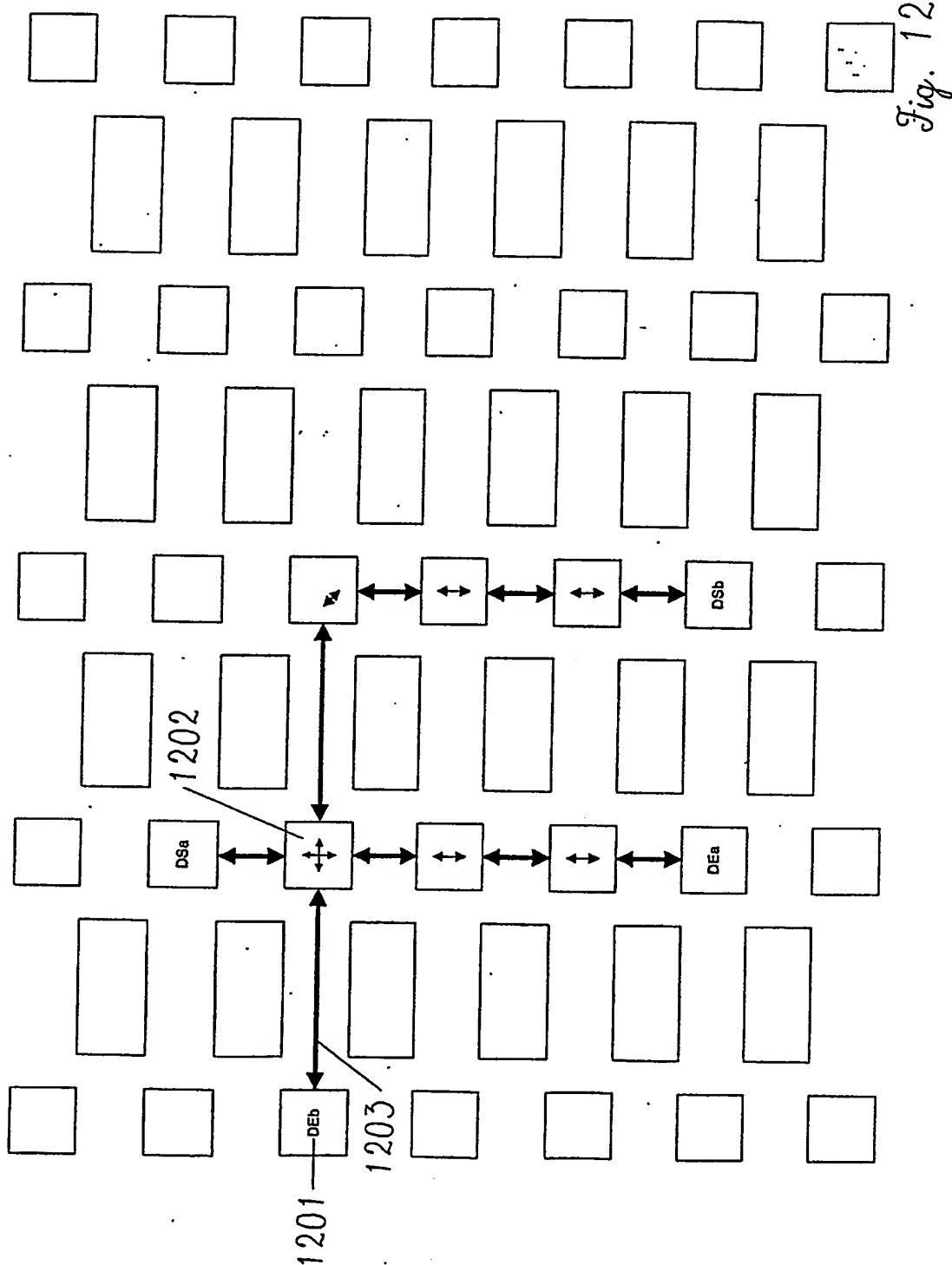
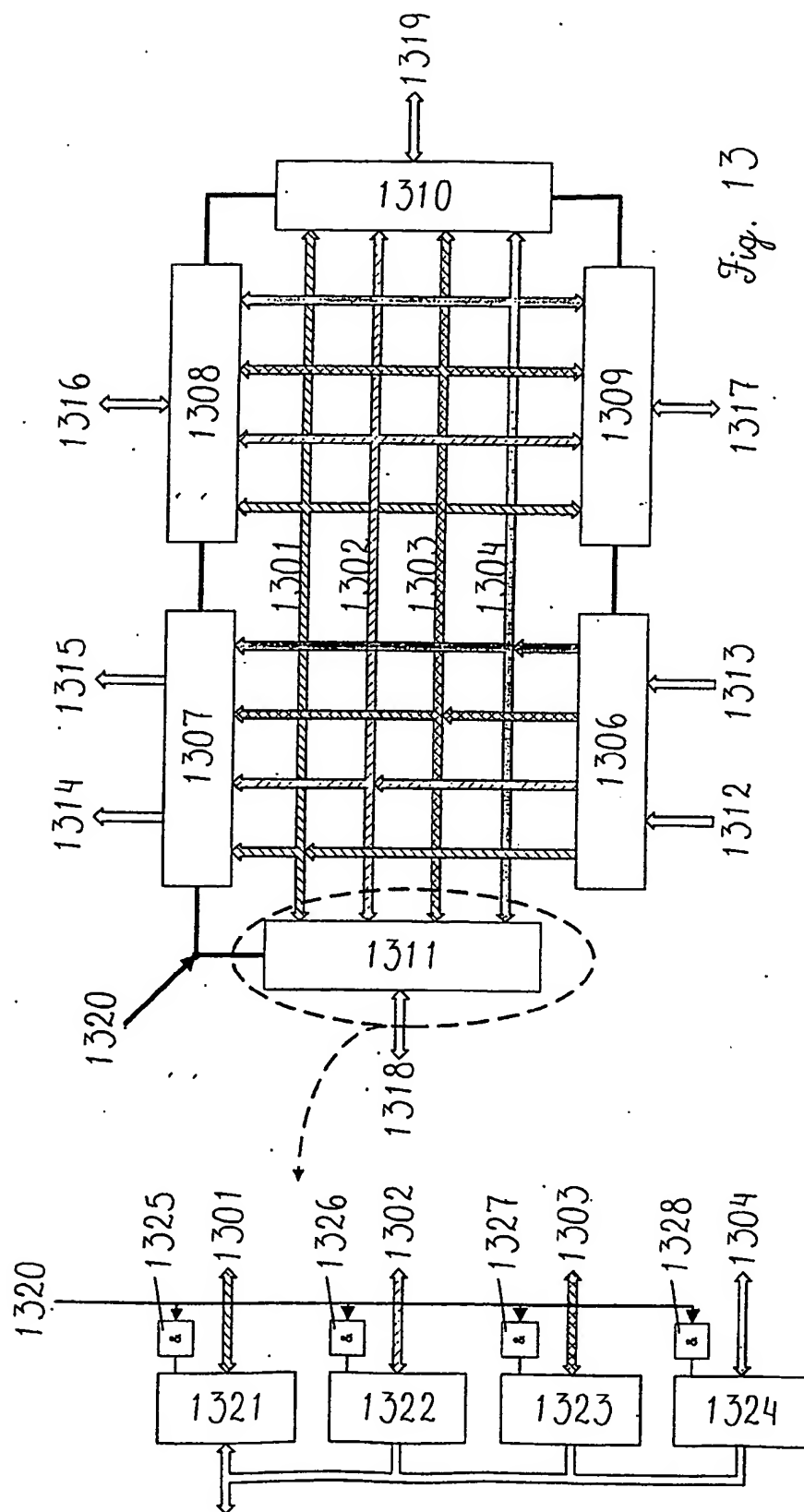
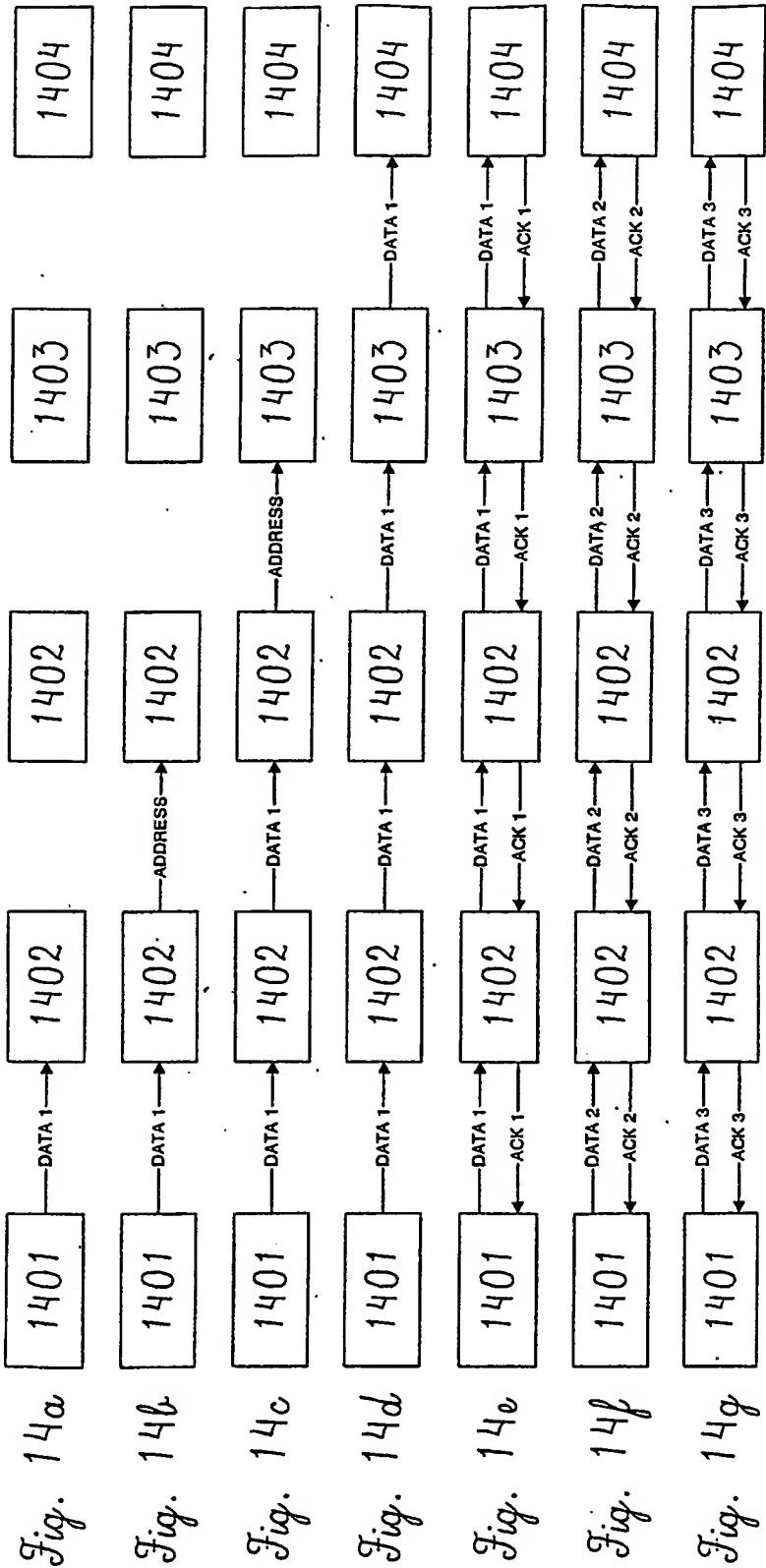


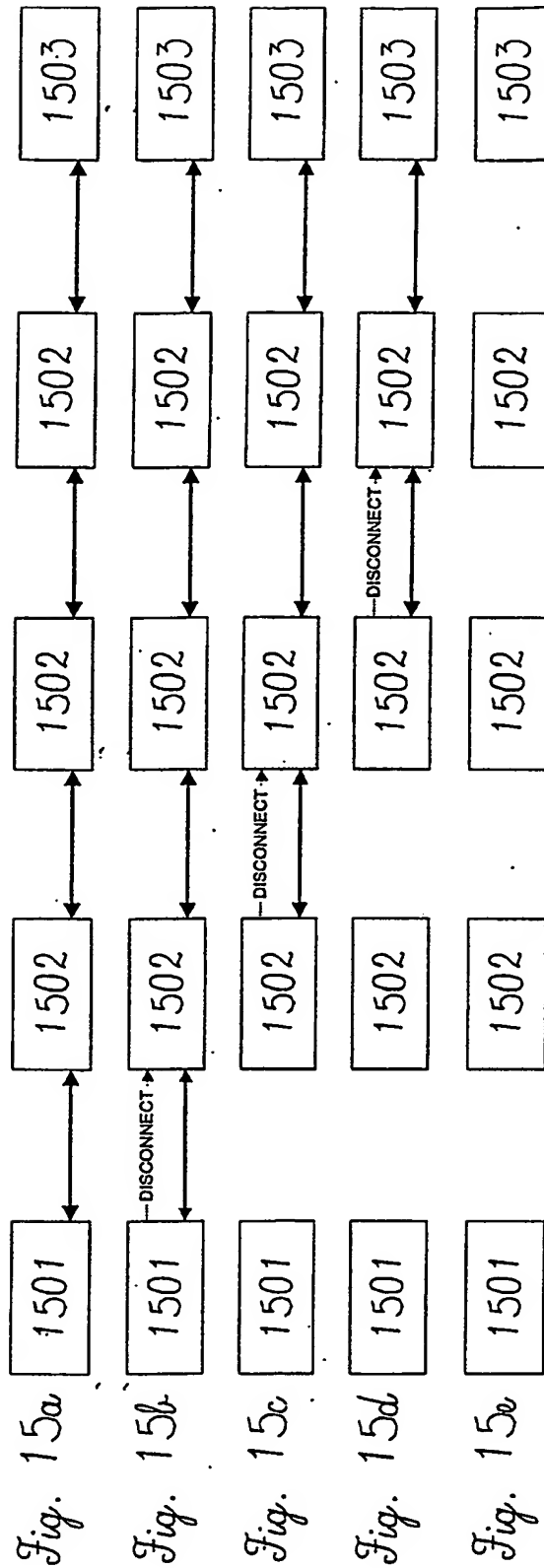
Fig. 10

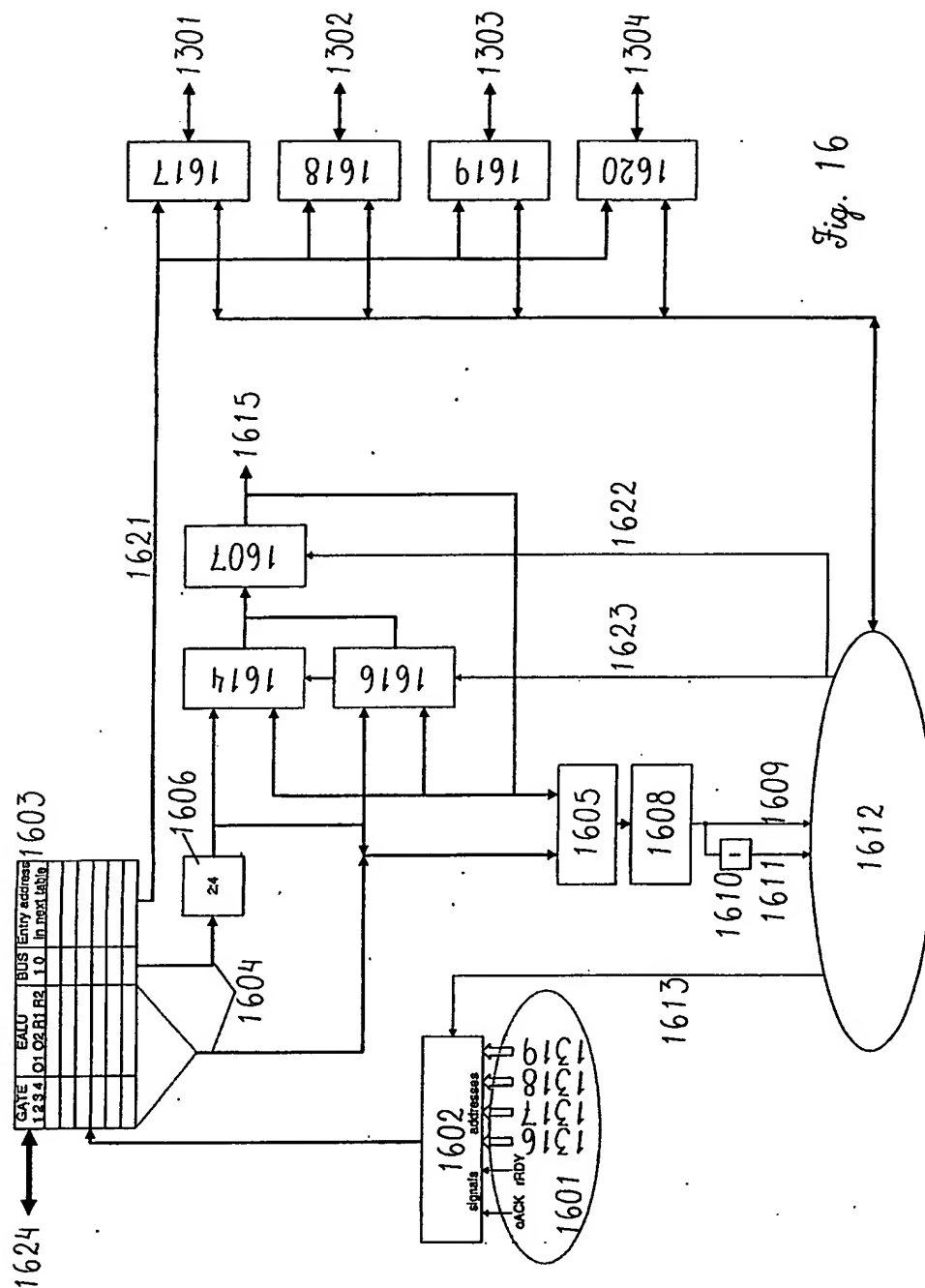












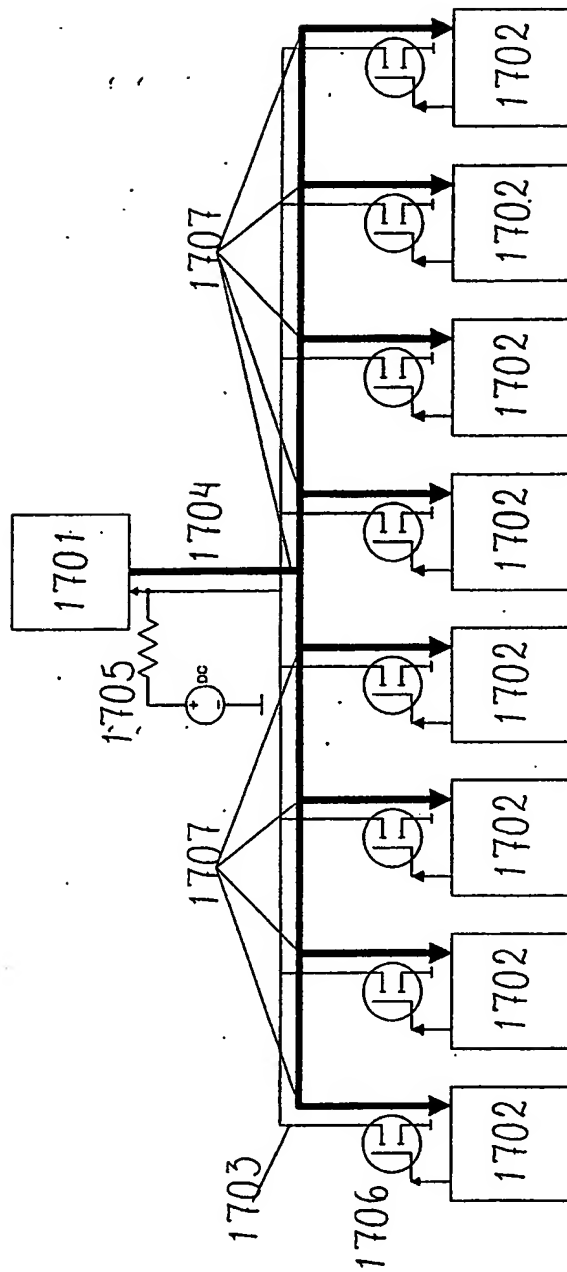


Fig. 17